(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 07253949 A

(43) Date of publication of application: 03.10.95

(51) Int. Cl G06F 15/16
G06F 7/72

(21) Application number: 05326008

(22) Date of filing: 30.11.93

(30) Priority: 30.11.92 IL 92 103921
16.02.93 IL 93 104753
06.09.93 IL 93 106923

(71) Applicant: FORTRESS U & T LTD

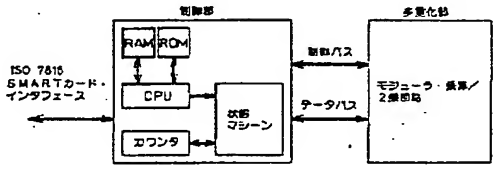
(72) Inventor: CRESSEL CARMI D
HENDEL DAVID
DROR ITAI
HADAD ISAAC
ARAZI BENJAMIN

(54) **MICRO ELECTRONIC DEVICE AND METHOD FOR EXECUTING MODULAR MULTIPLICATION AND MODULAR POWER**

(57) Abstract:

PURPOSE: To reduce the time required for executing modular multiplication, etc., based on the Montgomery method with respect to a micro electronic device and method for executing modular multiplication and power to large numbers.

CONSTITUTION: A micro electronic device is constituted of fractionized and switching-controllable compact synchronous micro electronic peripheral devices for a standard microprocessor having an appropriate clock means and control means and provided with a plurality of kinds of shift registers controlled by a clock means, two multiplexed serial/parallel mutiplexers, a borrow detector, an auxiliary subtractor, an auxiliary adder, a delay register, and a switching element. The electronic device is formed by integrating all of the above components so that the device can simultaneously and synchronously execute modular multiplication, modular square, and modular power.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平7－253949

(43)公開日　平成７年(1995)10月３日

(51)Int.Cl.[6]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ　15/16	３３０　Ｄ			
7/72				

審査請求　未請求　請求項の数28　ＦＤ　（全 34 頁）

(21)出願番号　特願平5－326008

(22)出願日　平成５年(1993)11月30日

(31)優先権主張番号　１０３９２１
(32)優先日　1992年11月30日
(33)優先権主張国　イスラエル（ＩＬ）
(31)優先権主張番号　１０４７５３
(32)優先日　1993年２月16日
(33)優先権主張国　イスラエル（ＩＬ）
(31)優先権主張番号　１０６９２３
(32)優先日　1993年９月６日
(33)優先権主張国　イスラエル（ＩＬ）

(71)出願人　593231656
フォートレス　ユー　アンド　ティー　リミティド
イスラエル国，ピアーシェバ　84110，ピー．オー．ボックス　844
(72)発明者　カーミ　デビッド　グレッセル
イスラエル国，モービル　ポスト　ネゲブ　85530，キブッツ　ウリム（番地なし）
(72)発明者　デビッド　ヘンデル
イスラエル国，ラーナナ，ハシャロン　ストリート　16
(74)代理人　弁理士　宇井　正一　（外４名）

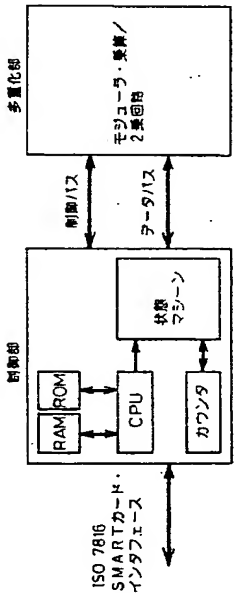
最終頁に続く

(54)【発明の名称】　モジュラ・乗算およびモジュラ・べき乗を遂行する超小形電子系装置ならびにその遂行方法

(57)【要約】

【目的】　本発明は、大きな数に対するモジュール・乗算およびべき乗を遂行するための超小形電子系装置ならびにその遂行方法に関し、モントゴメリの手法に基づきモジュラ・べき乗等の遂行に要求される時間を減らすことを目的とする。

【構成】　本発明は、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなり、各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタと、多重化され、かつ、直列／並列形の２つのみのマルチプレクサと、ボロー検出器と、補助的な減算器および加算器と、ディレイ・レジスタおよび切替素子とを備えており、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前述のすべての構成部品を集積化して形成する。



1

【特許請求の範囲】

【請求項１】　大きな数に対しモジュラ・乗算およびモジュラ・べき乗を遂行するための超小形電子系装置であって、

該超小形電子系装置は、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなり、

さらに、該超小形電子系装置は、

各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタ（Ｂ、ＳおよびＮ）と、

多重化され、かつ、直列／並列形の２つのみのマルチプレクサと、

ボロー検出器と、

補助的な減算器および加算器と、

ディレイ・レジスタおよび切替素子とを備えており、

前記超小形電子系装置は、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前記のすべての構成部品を集積化して形成することを特徴とする超小形電子系装置。

【請求項２】　前記超小形電子系装置が、ハードウェアの乗算、２乗およびべき乗に対し設計されたモントゴメリの方法をもとに展開されるような新奇かつ複合形で同期式のハードウェア装置により実現される請求項１記載の装置。

【請求項３】　前記超小形電子系装置が、モントゴメリの方法を展開することにより、並列動作方式に直列動作方式を取り入れた多数の同時処理と直列処理との複合形、すなわち、乗算、減算、加算、記憶形ディレイおよび2^kによる除算を遂行する装置として機能する請求項１記載の装置。

【請求項４】　前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗のための多数の直列処理を遂行し、かつ、膨大な内部バスの使用を回避する請求項１記載の装置。

【請求項５】　前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗のための多数の直列処理を遂行し、

前記超小形電子系装置は、一般の１μｍ技術を用いたＳＭＡＲＴカード用のＩＳＯ７８１６の標準規格により規定されるマイクロチップ上に形成される程度に充分コンパクトである請求項１記載の装置。

【請求項６】　前記超小形電子系装置が、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗のための多数の直列処理を遂行し、

前記超小形電子系装置は、基本のアーキテクチュアを変えることなく、特に、デュアルポート・アクセスのため

2

のメモリを再設計することなく、かつ、ファームウェアの要求が少ない状態で、１つの内部バスを備えた任意のマイクロプロセッサにより制御することが可能である請求項１記載の装置。

【請求項７】　前記超小形電子系装置が、マイクロプロセッサを使用してカスケード形のρ領域内での２乗および乗算の処理手順を規定し、

さらに、前記超小形電子系装置は、ｎビット長のシフトレジスタを含み、かつ、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗を遂行する多重化部を備え、該多重化部内に指数Ｅを記憶することが不要であるために、該多重化部による制御を簡単にし、また一方で、ほんのわずかな付加的なマイクロコントローラのＲＯＭコードのみを必要とする請求項１記載の装置。

【請求項８】　Ｂのレジスタが回転動作を行っている間に、オンザフライ方式により２乗の被乗数を用いてA_iのレジスタをロードする結果として、前記A_iのレジスタを前記Ｂのレジスタにより再ロードする際に、マイクロコントローラによりＢおよび／またはＢ－Ｎの前回の計算処理の最終値が取り出されるおそれが回避され、

このために、該マイクロコントローラのＲＡＭが節減され、かつ、２乗の繰り返し動作の各々において少なくともｎクロック分の実効的なクロック・サイクルを除去することが可能になる請求項１記載の装置。

【請求項９】　Ｚ／2^kがＮよりも大きいか、またはＮに等しいかを決定し、たった１回の直列形の減算のみ行うような比較的小さいＮオペランドを達成することにより、モントゴメリの方法による簡単な装置から、２つの記憶用レジスタおよび独立の直列形の減算処理が除去されると共に、Ｚ／2^k－Ｎに関する単一の直列形の検出を行うことが可能になる請求項１記載の装置。

【請求項１０】　３つの同時乗算処理を遂行する際に２つの直列／並列形の乗算器のみが使用されるように、半並列形式で回路の同期をとり、

このために、シリコンを用いた装置において全シリコン領域に対し直列／並列形の乗算器の占める面積の割合が４０％に抑えられ、

３つの直列／並列形の乗算器の代わりに２つの乗算器のみを使用することにより、シリコン領域内の直列／並列形の乗算器のセルを二重化し、

該二重化構造の乗算器のセルを動作させることにより、５１２ビットの乗算処理に必要な時間を従来の４５％に節減することが可能になる請求項１記載の装置。

【請求項１１】　ｋビットのシフトレジスタからなる１つのディジタルのディレイ素子を使用してＸの直列形の加算と乗算器（ＭＬ１）の直列結果との同期をとり、直列／並列形の乗算器の積または繰り返し処理が二重に記憶されることを防止する請求項１記載の装置。

【請求項１２】　各々がｋビットのシフトレジスタからなる２つのディジタルのディレイ素子を使用して３つの

直列形の乗算を遂行し、該乗算は、Ｎを１つの因子としたときにＢ・A_i、Ｘ・J_0および　Y_0・Ｎのように表される請求項１記載の装置。

【請求項１３】　処理の流れの中で２つの独立した乗算動作、すなわち、Ｘ・J_0および　Y_0・Ｎを遂行することができるように、１つのディジタルのディレイ素子を使用して直列／並列形の乗算器（ＭＬ２）の乗算動作の同期をとる請求項１記載の装置。

【請求項１４】　前記シフトレジスタ（Ｂ、ＳおよびＮ）が、ｎビット長またはｎ／２ビット長で構成され、ｎ／２の長さのモジュールに対するべき乗が、ｎビット長のべき乗に対し必要であろうと思われる実効的なクロック・サイクル期間の１／８より幾分少ない時間で遂行される請求項１記載の装置。

【請求項１５】　オリジナルの検索因子Ｔで処理がなされる場合、全ＲＳＡ記号のべき乗処理におけるρ領域内での乗算動作の回数が、半分近くに減少する請求項１記載の装置。

【請求項１６】　必要に応じて前もって計算することを仮定した場合、オンザフライ方式によりＡのレジスタをロードし、かつ、オンザフライ方式によりＳのレジスタの内容の大きさを予測し、さらに、オンザフライ方式により一部のオペランドの同期をとることにより、ｎビットの数の乗算処理ρ（Ａ・Ｂ）Ｎが、実効的なｍ（ｎ＋２ｋ）クロック・サイクルで完全に遂行される請求項１記載の装置。

【請求項１７】　小規模のボロー検出回路が付加され、かつ、制御メカニズムに簡単な付加物が付加されているようなモントゴメリの乗算処理に対し使用されるものと同じ機器の同じレジスタを用い、第２のモードにおいてＨパラメータの計算を行う請求項１記載の装置。

【請求項１８】　ρ領域内での乗算またはべき乗が公知のクロック・サイクルの処理で遂行されるように、すべての副処理過程および処理過程が、予め定められたクロック・サイクル数でそれぞれ実行され、
このために、内部の条件設定用ブランチを使用することなく、カスケード形かつ自励式のカウント・メカニズムからなる簡単化された制御が可能になる請求項１記載の装置。

【請求項１９】　モジュラ・乗算を遂行するための方法であって、被乗数Ａ、乗数ＢおよびモジュロＮの各々が、ｋビット長のｍキャラクタから構成され、乗数Ｂはモジュロ Ｎよりも大きくない値であり、前記方法は、下記のステップを有しており、
第１のステップで、Ｈパラメータと、他のパラメータの少なくとも最下位のキャラクタJ_0とを前もって計算し、かつ、該キャラクタJ_0をｋビットのレジスタにロードし、
第２のステップで、前記乗数ＢおよびモジュロＮを、それぞれ対応するｎビット長のレジスタにロードし、ここで、ｎ＝ｍ・ｋのように表され、
第３のステップで、ｎビット長のレジスタＳのビット値をすべて０にし、第４のステップで、ｉ番目の繰り返し動作をｍ回遂行し、ここで、ｉは０からｍ－１までの数であり、さらに、ｉ番目の繰り返し動作の各々は、以下の動作を含み、
（ａ）前記被乗数Ａのｉ番目のキャラクタA_iを、A_iのレジスタ手段から、レジスタおよびラッチ手段から選定された記憶手段へ転送し、
（ｂ）Ｘ＝Ｓ（ｉ－１）＋Ａ（ｉ－１）＊Ｂにより表されるＸの値を生成し、ここで、Ｓ（ｉ－１）はＳの更新された値であり、Ｓの更新は、次のように定義され、
①乗算手段に対し、Ｂのレジスタを周期的に右方向へシフトし、
②直列形式でＢをA_iにより乗算し、
③前記モジュロＮを周期的に右方向へシフトし、
④Ｓ（ｉ－１）がＮよりも大きくない場合、（ｉ－１）番目の繰り返し動作の後にＳのレジスタに記憶される値をＳ（ｉ－１）の更新された値として決定し、Ｓ（ｉ－１）がＮよりも大きい場合、直列形式でＳ（ｉ－１）からＮを引くことにより得られる値をＳ（ｉ－１）の更新された値として決定し、さらに、この結果として得られるＳ（ｉ－１）の更新された値を設定し、
⑤Ｓのレジスタを周期的に右方向へシフトし、さらに、各ビット毎に、乗算Ａ（ｉ－１）＊ＢをＳ（ｉ－１）の更新された値に加算し、
（ｃ）Ｘ（X_0）の最下位のキャラクタをJ_0により乗算し、ＮおよびＸがｋクロック・サイクルだけ遅延されている間に、$X_0 * J_0 \bmod 2^k$の値をY_0のレジスタ手段に入れ、
（ｄ）$Z = X + Y_0 * N$のＺの値を計算し、この計算は、次のように行われ、
①Ｎのレジスタに対し遅延かつ右方向へのシフトがなされた状態でY_0をＮにより乗算し、同時に、この乗算結果に対し、前述の周期的な右方向へのシフトがなされ、
②Ｘを$Y_0 * N$の値に加算し、
（ｅ）Ｚの最下位のキャラクタを無視し、残りのキャラクタをＳのレジスタに入れ、このときに、最後の繰り返し動作以外は、$Z/2^k$を入れることになり、
（ｆ）前述と同様の方法によりＳ（ｉ－１）の更新された値を決定するために、各ビット毎に$Z/2^k$とＮとを比較し、
（ｇ）前記被乗数Ａのｉ番目のキャラクタA_iを、前記の動作期間において、Ａのレジスタ手段にロードし、
第５のステップで、最後（ｍ回目）の繰り返し動作においては、$Z/2^k$の最下位のキャラクタを無視し、残りのキャラクタを、Ｃ¥ρ（Ａ＊Ｂ）ＮとしてＢのレジスタに入れ、
第６のステップで、前記第３および第４のステップを繰り返し、ここで、ＣがＮよりも大きい場合には、Ｃまた

ビットを無視する工程、
9．該ビットEのそれぞれに付いて、0か1かに関係なく、上記で定義された二乗方法による工程4及び5の操作を実行する工程であって、該被乗数と該乗数とが共に該レジスタBから派生されるものであり、且つ該モントゴメリー乗算器に於ける連続する特性値が該レジスタBからレジスタAiに格納される工程、
10．若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程9の操作終了後、前記で定義された二乗方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベースA＊である工程、及び
11．べき指数Eの全てのビットに付いて、工程8～10の操作が実行された後に、レジスタBの内容を前記オリジナルベースAで付加的に乗算し、D¥＝A^E mod Nとしての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている事を特徴とするモジュラ・べき乗D＝A^E mod Nを実行する方法。
【請求項28】　平均有効長がn／2ビットである2つの数値に付いて従来の乗算を実行する方法で有って、該乗算方法は、該請求項19に於いて定義されている乗算方法により該数値に対してモジュラ・乗算処理を実行するもので有って、該モジュラ、N、は全てが“1’s”（fffffff.....fff）で構成されたnービット数で、J0から1に対応するものであり、被乗数をレジスタBに格納し且つ請求項1に於いて定義されている乗算方法に従ってAを取り扱うものであり、Nは、全て1によりプリローディングレジスタNの手段によるか、或いは一連の“ハード”1を出力する為のNを出力するマルチプレクサーをセットすることにより、該Nは全て1と成りうるものである事を特徴とする方法。
【発明の詳細な説明】
【0001】
【産業上の利用分野】本発明は、複数種の素数および複合形素数モジュールからなるガロアの領域において、大きな数に対しモジュラ処理を遂行するための手法に関する。さらに詳しくいえば、本発明は、大きな数に対するモジュール・乗法およびモジュール・べき乗を遂行するための超小形電子系装置ならびにその遂行方法について言及するものである。このようなモジュール・乗算およびモジュール・べき乗は、公開キー暗号の認証や暗号化プロトコル（Protocol）に不可欠な演算を遂行する場合に適している。この種の演算は、小規模のマイクロプロセッサによっては通常の処理時間内に実行することが不可能である。
【0002】
【従来の技術、および、発明が解決しようとする課題】本発明は、インタリービング方式によるモントゴメリ（Montgomery）の多倍精度の乗算方法」として知られている手続きを、ハードウェアによって遂行することに関係する。このモントゴメリの乗算方法は、暗号化のソフトウェア指向システムに度々使用される。ここでは、モジュラ・べき乗を促進するために、唯一のオリジナルな方法が提供される。さらに、このモジュラ・べき乗を遂行するためのアーキテクチュアを簡単化し、かつ、通常使用されるような数の領域内でモジュラ・べき乗を遂行する装置の適用範囲を拡大するために、欠くことのできないプルーフが使用される。
【0003】上記の手続きに関する基本的な処理は、モントゴメリの方法論に基づきモジュラ・乗算を遂行するための手法に関連するような3つの公知の方法のいずれか一つにより実行される。これらの公知の方法の1つめは、P．L．モントゴメリ著の「試行除算を行わない方式のモジュラ・乗算（Modular Multiplication without trial division）」｛計算の数学（Mathematics of Computation）、第44巻、519～521頁、1985年発行｝に記載されている。なお、これ以降は、上記の1つめの方法は、単にモントゴメリの方法とよぶこととする。公知の方法の2つめは、S．R．デュッセ（Dusse）およびB．S．カリスキー　ジュニア（Kaliski Jr.）著の「モトローラ社製のDSP56000に向けられた暗号のライブラリ（A Cryptographic Library for the Motorola DSP 56000）」｛90年代の欧州暗号化の議事録（Proc. Eurocrypt　'90）、1990年にベルリンのシュプリンガ出版社（Springer-Verlag）により発行｝に記載されている。なお、これ以降は、上記の2つめの方法は、単にデュッセの方法とよぶこととする。
【0004】上記の手続きをハードウェアにより遂行する場合、機密保護機構や、オンザフライ（On the Fly）式の加算、減算、および、けた移動が付加される。さらに、総合的な出力が不適切であるような処理が除去される。さらにまた、シリコンを利用した設計に基づき、比較的容易に遂行できるような手段が開発されて集積化される。この集積化により実現された装置は、実際には、8ビット、16ビットまたは32ビットの中央処理装置（CPU）の従装置として、内部のデータ／アドレスバスに付加される。
【0005】本発明に関係する乗算／2乗機器は、簡単な同期形のけた送り式の設計がなされているために、これまでに達成された速度の何倍ものクロック速度で動作することが可能である。このクロック速度は、ボード上に実装されかつ不揮発性の記憶装置（メモリ）を支援するCPUにより実現される。乗算／2乗機器を用いた手法は、CPUのメモリのアーキテクチュアにおける設計変更を必要としない。この種のアーキテクチュアは、例えばフィリップ回路のように、並列形の乗算器とデュアルポートのメモリを用いることによって大きな数に対する高速のモジュラ・乗算を遂行する場合に規定されるものである。このモジュラ・乗算を遂行するための複数の

はC－NがBにとって代わり、さらに、HがAにとって代わることによってP＝ρ（C＊H）Nを計算し、
第7のステップで、最後の繰り返し動作により得られるPの値を、A＊BmodNと仮定することを特徴とする方法。

【請求項20】　nが、256と512との間の数から選定されるか、または、kの倍数の増分から選定される請求項19記載の方法。

【請求項21】　前記被乗数Aと前記乗数Bとが同じ数である場合に、モジュラ・2乗およびモジュラ・乗算を遂行する請求項19記載の方法。

【請求項22】　D＝A^E modNにより表されるモジュラ・乗算およびモジュラ・べき乗を遂行するための方法であって、該方法は、複数の乗算処理および2乗処理を含む請求項19記載の方法。

【請求項23】　1．モジュラスをレジスタNに格納する工程、
2．レジスタSを0にセットする工程、
3．べき乗化されるべきベースAをレジスタBに格納する工程、
4．べき指数Eをコンピューターのレジスタに格納する工程、
5．該べき指数Eを左にシフトさせる工程、
6．第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、
7．該ビットのそれぞれに付いて、0か1かに関係なく、上記で定義された乗算方法により該レジスタBの内容を二乗すると同時に、該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、
8．若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後該レジスタBの内容を該ベースAで乗算する工程、及び
9．べき指数Eの全てのビットに付いて、工程6～8の操作が実行された後に、D¥＝A^E modNとしての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている請求項22に記載の方法。

【請求項24】　1．モジュラスをレジスタNに格納する工程、
2．レジスタSを0にセットする工程、
3．べき乗化されるべきベースAをレジスタBに格納する工程、
4．べき指数Eをコンピューターのレジスタに格納すると共に、以下に定義する事前演算パラメータTをコンピューターCPUに格納する工程、記載の方法。
5．該べき指数Eを左にシフトさせる工程、
6．第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、
7．該ビットのそれぞれに付いて、0か1かに関係なく、請求項1で定義された乗算方法に関して工程4及び5を実行すると同時に、被乗数と乗数とがベースAであり、且つ該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、
8．若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後、請求項1で定義された乗算方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベースAである工程、及び
9．べき指数Eの全てのビットに付いて、工程7～8の操作が実行された後に、レジスタBの内容を前記パラメータで付加的に乗算し、TD＝A^E modNとしての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている事を特徴とする請求項19による繰り返し操作を実行することにより、モジュラ・べき乗D¥＝A^E modNを実行する方法。

【請求項25】　コンピューターCPUと乗算回路を含む制御手段から構成されている請求項19記載の方法により、モジュラ・乗算を実行する装置であって、該乗算回路は、乗数としてのn－ビットシフトレジスタB、モジュラスとしてのn－ビットシフトレジスタN、本発明に於いて定義されている値Sとしてのn－ビットシフトレジスタN、被乗数としてのk－ビットシフトレジスターAi、本発明に於いて定義されている値J0及びY0としてのk－ビットレジスタ手段、該レジスタBの内容と該レジスタAiの内容とを掛け合わせる乗算手段、付加的なn－ビット乗算器手段及び加算、減算、多重化及び遅延手段とを含んでいる事を特徴とする装置。

【請求項26】　該n－ビットレジスタとその他の構成部分との間の接続、及びラッチ回路以外の構成部分間の接続は1ビット接続である事を特徴とする請求項25記載の装置。

【請求項27】　1．べき指数Eをコンピューターの記憶手段に格納する工程、
2．モジュラスを前記レジスタNに格納する工程、
3．前記レジスタSを0に設定する工程、
4．前記した特許出願番号104753に記載された方法に従って、
$A^*=\rho(AH)_N$ の乗算操作を実行する工程、
（此処で、Aは、べき乗化されるべきオペランドであり Hは、前記で定義した事前演算パラメータである。）
5．該A＊を該ベースレジスタBに格納する工程、
6．該ベースレジスタBの内容に対して二乗演算操作を実行する工程、
7．該べき指数Eを左にシフトさせる工程、
8．第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第9及び10の操作を実行する為のビットに続く全てに対して、第1番目の1

手法の３つめは、フィリップ社製の電子部品「８３Ｃ８５２（条件付きのアクセス・アプリケーション用の機密保持機能付きの８ビット・マイクロコントローラ）」〔１９９０年８月にアインホーヘン（Einhoven）にて発表〕により実行される方法である。なお、これ以降は、上記の３つめの方法は、単にフィリップの方法とよぶこととする。

【０００６】上記のような基本的なアーキテクチュアは、任意のマイクロコントローラの設計に際し集積化が可能であり、かつ、メモリへのマッピングが可能であるような機器に適用される。さらに、この種の機器はまた、コマンドおよびオペランドを常時ロードし、かつ、最終的な応答結果を取り出して転送することが必要なマイクロコントローラと並列に動作することも可能でなければならない。

【０００７】このような要求に対する新奇な解は、２つの直列／並列式の乗算器のみを用いると共に、完全な直列形のパイプライン方式のアプローチを採用することである。このようなパイプライン方式のアプローチを採用することにより、シリコンの面積を節減することが可能になる。現在一般に用いられている周知の技術によれば、メモリ付きのマイクロコントローラを有すると共に上記の解を完全に満足するような装置を、４×４．５×０．２の大きさの電子系回路上に集積化することが可能である。このようにして得られる電子系回路は、ＩＳＯ７８１６の規格を満足するものである。ここで、ＩＳＯとは、国際標準化機構（International Organization for Standardization）の略号であり、認証カード、すなわち、集積回路カード（ＩＣカード）において特定される。上記ＩＳＯの中のＩＳＯ７８１６は、下記の３つの部分から構成される。

【０００８】①第１部…ＩＳＯ７８１６－１（物理特性）、１９８７年制定
②第２部…ＩＳＯ７８１６－２（接点の位置の寸法）、１９８８年制定
③第３部…ＩＳＯ／ＩＥＣ７８１６－３（電子信号および通信プロトコル）、１９８９年制定
なお、これ以降は、これらの３つの部分を併せてＩＳＯ７８１６とよぶこととする。

【０００９】本発明は、モントゴメリにより開示された数学的な革新事項に基づいて上記の新奇な解のアーキテクチュアを実現することに向けられている。本発明では、後述するように、モジュラ・べき乗の遂行に要求される時間を、公知の処理方法およびモントゴメリの方法を用いた場合に必要な処理時間の半分の時間とほとんど変わらない値にまで減らすために、幾つかの変形や、改良や、機能的な方法が提供される。

【００１０】ここで、本発明のモジュラ・乗算およびモジュラ・べき乗を遂行する装置および方法を述べる前に、一般的な計算の数学について大まかに説明することとする。

数学上の定義、一般的な法則および処理方法

素数および複合形基本モジュールからなる数の領域において、我々は、ＡおよびＢを、それぞれ被乗数および乗数として定義する。さらに、通常は、ＡまたはＢよりも大きい数としてＮを定義する。ただし、場合によっては、Ｎは、Ａよりも小さい数になり得る。さらに、Ａ、ＢおよびＮの各々を、ｍ×ｋ（積の記号は、×の代わりに、・または＊で表すこともある）＝ｎビット長のオペランドとして定義する。各ｋビットのグループは、キャラクタとよばれる。それゆえに、Ａ、ＢおよびＮの各々は、ｍキャラクタ長のオペランドから構成される。ここでは、モジュラ・乗算およびモジュラ・べき乗の最初の遂行過程の理解と、１ステップ毎にモジュラ・乗算およびモジュラ・べき乗を遂行する手続の説明を容易にするために、我々は、Ａ、ＢおよびＮの各々を、５１２ビット長（ｎ＝５１２）のオペランドとして定義する。さらに、ｋを３２ビット長に設定する。この３２ビットは、現在、乗算器のコストに見合ったｋの長さとみなせる。さらに、ｍ＝１６としてｍの値を設定する。このｍの値１６は、一つのオペランドにおけるキャラクタの数であり、かつ、５１２ビットのオペランドに対する２乗ループまたは乗算ループの繰り返しの回数である。この場合、明らかに、いずれのオペランドも整数である。

【００１１】さらに、我々は、モジュールの数の合同を表すために、記号“≡”を使用する。例えば、１６≡２ｍｏｄ７と記載されている場合、１６が２モジュロ７と合同であり、かつ、１６を７で割ったときの剰余が２であることを意味する。また、ＹｍｏｄＮ≡ＸｍｏｄＮと記載されている場合、ＸおよびＹの両方が、Ｎよりも大きい可能性がある。さらに、ＸおよびＹが正であるときは、それぞれの剰余は同一の値になるであろう。さらに、Ｙが負の整数である場合に、Ｙの合同はＹ＋ｕＮで表されることに注意すべきである。この場合、Ｙの合同がＮよりも小さい値であるならば、ｕは、Ｙの合同を正の値にするための最小の数として設定されるであろう。

【００１２】さらに、我々は、より限定された意味における合同を表すために、記号“￥”を使用する。ここで述べる処理過程においては、各種の値は、度々、望ましい値か、または、望ましい値とモジュールとの和になる。例えば、Ｘ￥２ｍｏｄ７と記載されている場合、Ｘは、２または９のいずれかに等しい。このときに、Ｘは、２ｍｏｄ７に対し限定された合同を有するものとして定義される。

【００１３】さらに、Ｘ＝ＡｍｏｄＮと記載されて場合、我々は、ＡをＮで割った場合の剰余としてＸを定義する。例えば、３＝４５ｍｏｄ６のように表される。数の理論においては、モジュラ・逆数が基本的な概念になる。例えば、Ｘのモジュラ逆数は、X^{-1}と表される。この場合、モジュラ逆数X^{-1}は、XX^{-1}ｍｏｄＮ＝１の関

係式により定義される。もし、Xの値が3に等しく（X＝3）、かつ、Nの値が13に等しければ（N＝13）、X^{-1}の値は9になる（$X^{-1}=9$）。すなわち、積3・9を13により割った値は1になる。

【0014】この場合、参照すべきビット、キャラクタおよび全オペランドの値の最上位または最下位を表示するために、頭字語MSおよびLSがそれぞれ使用されることがある。本明細書中で、Nは、値N、および、この値Nを含むシフトレジスタの名前の両方を意味している。AおよびNは、べき乗処理の全過程を通して一定の値である。さらに、Aは、べき乗処理がなされるべき数の値である。べき乗処理の最初の繰り返し動作においては、BはAに等しい。Aはまた、累算された値が存在するレジスタの名前でもある。この場合、累算された値は、最終的に、べき乗処理の望ましい結果に等しくなる。Sは、一時的な値を示すと共に、値Sに対し限定された合同（¥）の関係を有する値が記憶されるようなレジスタを示す。S（i－1）は、i回目の繰り返し動作の始まりにおけるSの値を意味する。S_0は、S（i）の値の最下位（LS）のキャラクタを示す。

【0015】ここで、我々は、ρ領域（この“ρ”はベクトルで表示すべきものであるが、電子出願の形式では、ρをベクトルにて表示することができないので、やむを得ず通常のギリシャ文字で表示することとする）における乗算ρ（A・B）Nの処理過程を簡単に説明する。なお、乗算ρ（A・B）Nの詳しい定義は、後ほど行うこととする。

【0016】このρ（A・B）N以外の記号は、算術計算の中で通常使用されるものである。

モントゴメリ方式のモジュラ・乗算

モジュラ・乗算A・BmodNを遂行するための古典的なアプローチにおいては、積A・Bの剰余は、除算処理＊を利用することにより計算される。しかしながら、このような除算動作を実行することは、乗算動作を実行することよりも難しい。

【0017】モントゴメリのモジュラ・縮小法を用いることにより、上記の除算処理は、実質的に、前もって計算された定数を使用した乗算処理に置き換えられる。モントゴメリの関数ρ（A・B）Nは、ρ領域内で積A・Bの乗算モジュロNを遂行する。ρ領域から通常のモジュールの領域への検索処理は、ρ（A・B）Nの結果と前もって計算された定数Hをもとにρを規定することにより遂行される。ここで、Pがρ（A・B）Nと合同である場合（P≡ρ（A・B）N）、ρ（A・B）NはA・BmodNに等しくなる（ρ（A・B）N＝A・BmodN）。それゆえに、ρ領域での2つの乗算処理によって通常のモジュラ・乗算がなされることになる。

【0018】効果的なモジュラ・縮小法を使用する意図は、nビット長および2nビット長のオペランドに対する一連の乗算および除算動作を回避することにある。このような乗算動作および除算動作の回避は、元の値がnビット長であってかつ最高値がnビット長の最終結果を生成するようなオペランドに対し一連の乗算、加算および減算を実行することにより実現される。上記のようなモントゴメリの指針を証明するために、我々は、所定のA、B、および奇数のN（この奇数のモジュールは、常に、単純な大きな素数かまたは複合形の大きな素数のいずれかである）に対し、次のようなQが最終的に存在することに注意すべきである。すなわち、A・B＋Q・Nが、最下位のnビットの値が0になるような数になるという条件を満足するQが存在することである。さらに詳しくは、このような条件を下記の式(1) に示す。

【0019】

【数1】

$$P \cdot 2^{n} = A \cdot B + Q \cdot N \qquad (1)$$

【0020】この式(1) は、最下位のnビットの値が0になるような2nビット長の表現が可能であることを意味する。ここで、$I \cdot 2^{n}$がImodNと合同である（$I \cdot 2^{n} \equiv I \bmod N$）と仮定する。この場合、Iはすべての奇数のNに対して存在する。前述の式(1)の両※辺にIを掛けることにより、式(1) の左辺では、下記の式(2) に示すような合同の関係が導き出される。

【0021】

【数2】

$$P \cdot I \cdot 2^{n} \equiv N ;\ (\text{ただし、} I \cdot 2^{n} \equiv 1 \bmod N) \qquad (2)$$

【0022】また一方で、式(1)の左辺では、下記の式(3) に示すような合同の関係が導き出される。★

【0023】

【数3】

$$A \cdot B \cdot I + Q \cdot N \cdot I \equiv AB \cdot I \bmod N ;\ (\text{ただし、} Q \cdot N \cdot I \equiv 0 \bmod N) \qquad (3)$$

【0024】この結果、前述の式(2) および式(3) より下記の式(4)が導き出される。☆

【0025】

【数4】

$$P \equiv A \cdot B \cdot I \bmod N \qquad (4)$$

【0026】残念なことではあるが、この式(4) より、ρ領域の乗算が実行される度に寄生因子（寄生関数）Iが導入されることがわかる。ここで、ρ演算子を下記の式(5) のように定義する。

【0027】

【数5】

15

【００４４】この式(13)により、定数Ｊが求まる。ここで、ＪはＮのみの関数なので、前もって計算された定数となる。さらに、明らかなことではあるが、我々は、Ｎよりも小さい正の値のＪを選定しなければならない。これまでの説明より当業者にとっては明らかなように、上記の処理過程において、所定のＡ、Ｂ、Ｎ、および、前もって計算された定数に対し、３つの乗算処理と、１つの加算処理と、最高の減算処理とを遂行することによってρ（Ａ・Ｂ）Ｎが得られる。さらに、このようにして得られた結果と、同じような処理過程と、前もって計算された定数Ｈ（モジュールＮの関数）とを用いることにより、Ａ・ＢｍｏｄＮが求められる。この場合、ＡはＢに等しくなるので、モジュラ・算術計算により２乗または乗算を行うための装置に対し、このような演算子を使用することが可能になる。

【００４５】インタリービング方式によるモントゴメリのモジュラ・乗算

16

＊前述のセクションにおいては、すべてｎビット長の複数のオペランド、および、２ｎ＋１ビットの記憶領域が要求される複数の計算結果に対して乗算を必要とするようなモジュラ・乗算の方法を述べてきた。ここで、さらに、モントゴメリのインタリービング方式による縮小法（既に記載済みのデュッセの論文に記述されている）を利用することにより、より短いオペランド、レジスタ、およびハードウェア乗算器を用いた乗算動作を実行することができる。この結果として、論理ゲートの数が比較的少ない電子装置による処理が可能になる。

【００４６】さらに、ｋビットの乗算器を用いることにより、ｋビット長のキャラクタを定義することが容易に行える。この場合、ｎビット中にｍ個のキャラクタが存在する（ｍ・ｋ＝ｎ）。Ｊの最下位の文字としてJ_0を定義することにより、下記の式(14)が導き出される。

【００４７】

【数１４】

$$J_0 \equiv -N_0^{-1} \bmod 2^k \quad (J_0\text{は、}N\text{が奇数のときに存在}) \qquad (14)$$

【００４８】ここで、前述のモントゴメリのインタリービング方式による縮小法を用い下記のステップ１）～ステップ５）を遂行することによって、次のような初期条件の下でｍ回の繰り返し動作の後にρ（Ａ・Ｂ）Ｎが規定される。本発明の回路は、これらの複数のステップを並列方式により実行する。

＊

※初期条件：Ｓ（０）＝０（最初（１回目）の繰り返し動作の始まりにおけるＳに対し限定された合同の関係を有する値）

【００４９】

【数１５】

For i = 1, 2…m :

1) $X = S(i-1) + A_{i-1} \cdot B$

{A_{i-1} はAの（ｉ－１）番目のキャラクタ；
Ｓ（ｉ－１）はｉ回目の繰り返し動作の始まりにおけるＳの値を示す}

value

2) $Y_0 = X_0 \cdot J_0 \bmod 2^k$

（積$X_0 \cdot J_0$中の最下位のｋビット）

3) $Z = X + Y_0 \cdot N$

4) $S(i) = Z / 2^k$

（Ｚ中の最下位のｋビットは常に０であり、それゆえにＺは常に2^kにより割り切れる）

5) $S(i) = S(i) \bmod N$

（Ｎは、Ｎよりも大きいＳ（ｉ）から引かれる）

最終的に、最後の繰り返し動作において、

$$C = S(m) = \rho(A \cdot B)N \qquad (15)$$

【００５０】ここで、ステップ５）の除算処理は、Ｚ中の最下位のｋビットが常に０である場合の右方向へのｋビットのけた移動（シフト動作）に相当する。または、除算処理回路に見られるように、Ｚ中の最下位のｋビットが単純に無視される。上記のように、最後の繰り返し動作の後に、式(15)が得られる。あるいは、必要に応じて、Ｎを引いた後に式(15)が得られる。Ｆ＝Ａ・ＢｍｏｄＮを導き出すために、我々は、ρ領域におけるρ（Ｃ・Ｈ）Ｎの計算を実行しなければならない。

【００５１】ここで、我々は、すべてのＳ（ｉ）に対し、Ｓ（ｉ）の値が２Ｎよりも小さいことを証明する（モントゴメリの証明には含まれていない）。ここでの

13 14

$$P \equiv A \cdot B \cdot I \bmod N \equiv \rho (A \cdot B) N \quad (5)$$

【００２８】さらに、式(5) のＰを、“ρ領域における　＊演算することにより遂行される。
ＡとＢとの乗算”とよぶこととする。ρ領域からの検索　【００２９】
処理は、下記の式(6) に示すように、Ｐ・Ｈに対しρを＊　【数６】

$$\rho (P \cdot H) N \equiv A \cdot B \bmod N ; \quad (6)$$

【００３０】この式(6) のような合同の関係におけるＰ　※【００３１】
を式(4) のＰで置き換えることにより、Ｈの値が導き出　【数７】
される。この経過を下記の式(7) に示す。　※

$$\rho (P \cdot H) N \equiv (A \cdot B \cdot I)(H)(I) \bmod N ; \quad (7)$$

（Ａ・Ｂ・Ｉ←Ｐ；　Ｈ←Ｈ；　Ｉ←すべての乗算動作は
寄生関数Ｉを発生する）

【００３２】ここで、Ｈが、I^2 の逆数と合同である場　★【００３３】
合、式(7) は有効となり、下記の式(8) が成立する。　★　【数８】

$$H = I^{-2} \bmod N \equiv 2^{2n} \bmod N \quad (8)$$

（ＨはＮの関数であり、Ｈパラメータとよばれる）

【００３４】Ａ・Ｂに対しρ演算子を規定するために、　☆(9) が成立する。
前もって計算された定数Ｊを用いて下記のステップ１）　【００３５】
からステップ５）までの処理を遂行することとする。こ　【数９】
の場合、最終的に、ステップ５）において、下記の式　☆

1) $X = A \cdot B$
2) $Y = (X \cdot J) \bmod 2^n$ （最下位のｎビットのみ必要）
3) $Z = X + Y \cdot N$
4) $S = Z / 2^n$ （Ｊに対する条件を満たすために、Ｚが2^nに
より割り切れるようにすることが必要となる）
5) $P \leftarrow S \bmod N$ （もしＳ≧Ｎであれば、ＳからＮが引かれる）

最終的に、ステップ５）において、

$$P \leftarrow \rho (A \cdot B) N \quad (9)$$

〔Ｎを引いた後、必要な場合には、Ｐ＝ρ（Ａ・Ｂ）Ｎ〕

【００３６】これらの処理に続き、下記の式(10)が導き　◆【００３７】
出される。　◆　【数１０】

$$Y = A \cdot B \cdot J \bmod 2^n \quad (10)$$

（ｎビット中の最下位ビットのみ使用）

【００３８】さらに、下記の式(11)が導き出される。　＊【数１１】
【００３９】　＊

$$Z = A \cdot B + (A \cdot B \cdot J \bmod 2^n) \cdot N \quad (11)$$

【００４０】ここで、Ｚが2^n（Ｚの最下位のｎビット　※【００４１】
が０でなければならない）により割り切れるためには、　【数１２】
下記の式(12)に示す合同が存在することが必要である。※

$$(A \cdot B + (A \cdot B \cdot J \bmod 2^n) \cdot N) \bmod 2^n \equiv 0 \quad (12)$$

【００４２】さらに、この式(12)のような合同が存在す　★ばならない。
るためには、Ｎ・Ｊ$\bmod 2^n$ が－１と合同であること　【００４３】
が必要である。すなわち、下記の式(13)が成立しなけれ★　【数１３】

$$J \equiv -N^{-1} \bmod 2^n \quad (13)$$

処理過程に使用されるオペランドに対し、下記の式(16)のような３つの不等式が成立することに注意すべきである。

【００５２】【数１６】

$$S(i-1)<N; \quad B<N \text{ and } A_{i-1}<2^k \tag{16}$$

【００５３】（これらの不等式中の最初の２つは、Ｓ（ｉ－１）およびＢがＮに等しいかまたは大きい場合に、繰り返し動作の始まりにおいて、これらのＳ（ｉ－１）およびＢからＮを引いたときに成立する。さらに、2^k が、最上位（ＭＳ）のビットが１であるようなｋ＋※１ビット長の数であり、かつ、A_{i-1} がｋビット長のオペランドである場合に、３つめの不等式が成立する。）定義により、上記の式(17)が成立する。

【００５４】【数１７】

$$S(i)=Z/2^k \quad \text{（減算が可能な最後の部分におけるＳの値）} \tag{17}$$

【００５５】前述の一揃いの式において置換を行うことにより、下記の式(18)が成立する。

【００５６】【数１８】

$$Z=S(i-1)+A_{i-1}\cdot B+(X_0\cdot J_0 \bmod 2^k)N \tag{18}$$

【００５７】ここで、式(18)における各要素の最高値を取り入れることにより、下記の式(19)のようなＺに関する不等式が成立する。

【００５８】【数１９】

$$Z<(N-1)+(2^k-1)\cdot(N-1)+(2^k-1)\cdot N$$
$$=2^k N+2^k N-N-2^k \tag{19}$$

【００５９】この不等式より、下記の式(20)が確実に成立する。

【００６０】【数２０】

$$Z<2^k\cdot N+2^k\cdot N \tag{20}$$

【００６１】ここで、式(20)の不等式の両辺を2^K で割ることにより、式(21)が得られる。

【００６２】【数２１】

$$Z/2^k<N+N \tag{21}$$

【００６３】この式(21)の不等式より、いかなる場合にも、Ｓ（ｉ）またはＢを調整するためには、Ｎを１回だけ減算すればよいことが証明された。

例１

インタリービング方式によるモジュラ・乗算

１６進法のモードの手動形計算機を使用することにより、インタリービング方式によるモジュラ・乗算を利用した計算の有効性が容易に証明される。まず初めに、１６進法のフォーマットを用いて次のように数を設定する。

【００６４】Ｎ≒ａ５９（モジュロ）、Ａ＝９９ｂ（乗数）、Ｂ＝５ｃ３（被乗数）、ｎ＝１２（Ｎのビット長）、ｋ＝４（乗数のビットの大きさであり、キャラクタの大きさでもある）、および、ｍ＝３（ｎ＝ｋ・ｍ）さらに、J_0 ＝７（７・９≡－１ｍｏｄ１６）、および、Ｈ≡2^{2x12}ｍｏｄａ５９≡４４ｂが設定される。

【００６５】ここで要求される結果は、Ｆ≡Ａ・ＢｍｏｄＮ≡９９ｂ・５ｃ３ｍｏｄａ５９≡３７５８１１ｍｏｄａ５９＝220_{16}である。インタリービング方式によるモジュラ・乗算を利用した計算の処理過程を下記に示す。

初期条件：Ｓ（０）＝０

【００６６】【数２２】

初期条件 S（0）＝0

ステップ1 $X=S(0)+A_0 \cdot B=0+b \cdot 5c3=3f61$

$Y_0=X_0 \cdot J_0 \bmod 2^4=7$

$Z=X+Y_0 \cdot N=3f61+7 \cdot a59=B7d0$

$S(1)=Z/2^4=87d$（Nよりも小さい）

ステップ2 $X=S(1)+A_1 \cdot B=87d+9 \cdot 5c3=3c58$

$Y_0=X_0 \cdot J_0 \bmod 2^4=8 \cdot 7 \bmod 2^4=8$

$Z=X+Y_0 \cdot N=3c58+52c8=8f20$

$S(2)=Z/2^4=8f2$（Nよりも小さい）

ステップ3 $X=S(2)+A_2 \cdot B=8f2+9 \cdot 5c3=3ccd$

$Y_0=d \cdot 7 \bmod 2^4=b$

$Z=X+Y_0 \cdot N=3ccd+b \cdot a59=aea0$

$S(3)=Z/2^4=aea$，$S(3)>N$，

$S(3)=aea-a59=91$

ゆえに、$C=\rho(A \cdot B)N=91_{16}$ (22)

【0067】 ＊ ＊【数23】

（ρ領域の検索は、ρ（C・H）Nを計算することにより遂行される：ここで、再び初期条件S（0）＝0を設定する）

ステップ1 $X=S(0)+C_0 \cdot H=0+1 \cdot 44b=44b$

$Y_0=d$

$Z=X+Y_0 \cdot N=44b+8685=8ad0$

$S(1)=Z/2^4=8ad$

ステップ2 $X=S(1)+C_1 \cdot H=8ad+9 \cdot 44b=2f50$

$Y_0=0$

$Z=X+Y_0 \cdot N=2f50+0=2f50$

$S(2)=Z/2^4=2f5$

ステップ3 $X=S(2)+C_2 \cdot H=2f5+0 \cdot 44b=2f5$

$Y_0=3$

$Z=X+Y_0 \cdot N=2f5+3 \cdot a59=2200$

$S(3)=Z/2^4=220_{16}$

【0068】最終的に得られた値は、99b・5c3mod a59であり、前述の要求される結果に一致する。上記の乗算動作の有効性は、各ステップにおいて最下位の0のkビットを無視する場合には、基本的に、上位のnビットに対し2^kを掛けることになることを我々が認識したときに、直観的に理解することができる。さらに、各ステップにおいて、乗数のi番目の部分は、2^{ik}により乗算される数である。このような乗算処理により、上記の部分は、S（i）と同じランクになる。

【0069】モントゴメリの機器における1つの乗算処理過程内のモジュラ・縮小法

例えば、NISTのディジタル記号の標準化や、中国式の剰余定理を用いたモジュラ・べき乗のような多くの暗号化過程では、第2のモジュロよりも大きい（大抵の場合、2倍よりも大きい）数を減らすことが要求される。これらのモジュラ・縮小法は、インタリービング方式による1つのモジュラ・乗算処理過程において効果的に実行され得る。このモジュラ・乗算処理過程は、本発明に

初期条件：S（0）＝0，A＝t＝0af59b，
B＝R＝141d，N＝q＝2b13
ステップ1　X＝S（0）＋A$_0$・B＝0＋9b・141d＝c2d8f
Y$_c$＝X$_0$・J$_0$ mod 2^t＝8f・e5 mod 2^8＝eb
Z＝X＋Y$_0$・N＝c2d8f＋eb・2b13
＝33b800
S（1）＝Z／2^t modN＝33b8（Nよりも大きい）
S（1）＝33b8－2b13＝8a5
ステップ2　X＝S（1）＋A$_1$・B＝8a5＋f5・141d
＝134866
Y$_0$＝X$_0$・J$_0$ mod 2^t＝66・E5 mod 2^8＝3e
Z＝X＋Y$_0$・N＝134866＋3e・2b13
＝1db700
S（2）＝Z／2^t modN＝1db7
ステップ3　X＝S（2）＋A$_2$・B＝1db7＋0A・141d
＝e6d9
Y$_0$＝d9・e5 mod 2^8＝1d
Z＝X＋Y$_0$・N＝e6d9＋1d・2b13＝5c800
S（3）＝Z／2^t modN＝5c8

【００７７】最終的に、前述のように、ｔｍｏｄｑが５ｃ８に等しいことが確認される。

べき乗

ここでは、Ｄ．ヌース（Nuth）著による「コンピュータ・プログラミングの技術（The Art of Computer Programming）」｛半数学的アルゴリズム（Seminumerical Algorithms）、第２巻、アディソンーウェスリ（Addison-Wesley）社、読書協会（Reading Mass）、１９８１年発行｝による処理手順の方法をもとに、モジュラ・べき乗を遂行するための２乗および乗算の処理手順を説明する＊る。なお、これ以降は、上記の方法をヌースの方法とよぶこととする。

【００７８】まず初めに、我々は、前のセクションにおいて定数を予め計算していると仮定する。我々はまた、本発明の装置が、ρ領域内で２乗および乗算の両方を遂行することができると仮定する。このときに、我々は下記の式(26)のような計算を行うこととする。

【００７９】

【数２６】

$$C = A^E \bmod N \qquad (26)$$

【００８０】この式(26)の計算処理過程を下記に示す。ここで、Ｅ（ｉ）は、指数Ｅに対し２進法のビット表示を行ったときのｉ番目のビットを示す。このｉ番目のビットは、インデックスが１の最上位のビットで始まり、インデックスがｑの最下位のビットで終わる。

【００８１】

【数２７】

係る機器や、モントゴメリのアルゴリズムへの機能的な拡張を利用している。

【００７０】前述の幾つかの例では、モジュロの長さであるようなオペランドのｎビットが、Ｎの正確な長さでもあることが暗に示されている。このような関係は、通常のべき乗および乗算に対しては、ひじょうに効果的である。しかしながら、数の大きさの縮小が必要な場合には、第２の定数$I^{-1}=2^n \bmod N$を使用することが有効である。この第２の定数は、所定の数により乗算されるモントゴメリの数値を縮小することが要求される場合、１つの乗算処理の規模を最小のレベルにまで縮小するものである。上記の定数I^{-1}は、定数Ｈを計算する場合（Ｈパラメータの計算に関するセクションを参照のこと）と同様のメカニズムによって計算される。さらに詳しくいえば、このI^{-1}の計算は、除数レジスタの最上位＊

＊のビットにおいて１のビット値が存在するように、除数のオペランドの最上位の部分にモジュールＮを配置することにより実行される。ここで、けた移動／試行減算の回数は、明らかにｎ÷Ｉ－Ｌでなければならない。ただし、Ｌは、Ｎが関係するビットの数である。この場合、I^{-1}は、Ｌビット長のオペランドになることに注意すべきである。

【００７１】上記の前提条件を証明するために、まず初めに、我々は、Ａ・Ｂ$\bmod$Ｎ（ρ（Ａ・Ｂ）Ｎ）に対するモントゴメリの乗算処理によってＡ・Ｂ・Ｉ$\bmod$Ｎと合同の関係を有するものが生成されることを繰り返し述べる。ＢがI^{-1}に等しい（$B=I^{-1}$）と仮定した場合、下記の式(24)が成立する。

【００７２】

【数２４】

$$\rho(A\cdot I^{-1})N=A\cdot I^{-1}\cdot I \bmod N=A \bmod N \qquad (24)$$

【００７３】例２

インタリービング方式によるモントゴメリの縮小法

ｔがモジュロｑに縮小され得ること（ｔ　ｍｏｄｑ）を証明するために、乗数レジスタの長さが、ｑの長さよりも大きくなるように設定する。ここで、最初に記憶されるｔは、２４ビットの長さを有する。

【００７４】さらに、ワードの長さ（機器の乗数の大きさ）が８ビットであるとし、かつ、次のような変数を仮定する。

ｎ＝２４、ｋ＝８、ｔ＝０ａｆ５９ｂ、ｑ＝２ｂ１３、および、$R=I^{-1}=2^{24} \bmod q=141d$

単純な除算の計算処理を用いて比較を行うことにより、ｔｍｏｄｑが５ｃ８に等しいこと（ｔｍｏｄｑ＝５ｃ８）がわかる。このような処理過程を下記に示す。

【００７５】この場合、一回のモントゴメリの乗算処理において縮小および検索が遂行されることに注意すべきである。

【００７６】

【数２５】

```
a)  B＝A
    FOR j＝2 TO q
a)    B≒ρ（B・B）N
b)    B≒ρ（B・H）N（上記のステップa），b）は、
                    B≒B² mod Nに相当する）
      IF  E（j）＝1  THEN
a)'   B≒ρ（B・A）N
b)'   B≒ρ（B・H）N（上記のステップa）'，b）'は、
                    B≒B・A mod Nに相当する）
```

【００８２】各ステップから次のステップに移行する際に、ＢがＮに等しいかまたは大きい場合には、常に、ＢからＮが引かれる。最後の繰り返し動作の後に、Ｂの値は、A^E ｍｏｄＮに対し限定された合同の関係を有するようになる（Ｂ≒A^E ｍｏｄＮ）。本発明の回路を用いてモジュラ・べき乗を遂行する場合に、より効果的に利用され得るような特許権を有する複数のプロトコルが存在する。ここで、我々は、本発明で述べる方法において、通常のべき乗処理の速度が２倍になるような２つの暗号化プロトコルを挙げることとする。

【００８３】１つめの暗号化プロトコルは、Ｒ．Ｌ．リベスト（Rivest）他著の「ディジタル記号および公開キー暗号化システム（A Method for Obtaining Digital Signatures and Public Key Cryptosystems）」（ＡＣＭ委員会（Comm. of the ACM）、第２１巻、１２０～１２６頁、１９７８年発行）に記載されている。なお、これ以降は、上記の１つめのプロトコルを、ＲＳＡの方法とよぶこととする。２つめの暗号化プロトコルは、W. ディッフィ（Diffie）およびM. Ｅ．ヘルマン（Hellman）著の「暗号手法に関する新しい指針（New Directions in Cryptography）」（情報理論のＩＥＥＥ議事録（IEEE Trans. on Inform. Theory）、ＶＯＬ．ＩＴ－２２、６４４～６５４頁、１９７６年発行）に記載されている。なお、これ以降は、上記の２つめのプロトコルを、ディッフィーヘルマンの方法とよぶこととする。これらの２つの方法においては、一定の指数を用いて大部＊分の難しいべき乗が遂行される。

【００８４】次のセクションで述べる方法（ρ領域からの検索のために効果的な方法）は、一定の指数を使用することによって計算処理に必要な計算時間を減らすことに言及する。この方法が用いられた場合、前述のステップｂ）'のべき乗処理過程（すべてのρ（Ａ・Ｂ）Ｎの乗算処理）が取り除かれる。さらに、べき乗処理に関するｑ回目の繰り返し動作の後に得られるＢの最終値が、モントゴメリのρ領域において、前もって計算された定数Ｔにより乗算される。

【００８５】上記の暗号化プロトコルに従事する人々にとっては、中国式の剰余定理（前述のヌースの文献中に記述されている）を用いた回路によってＲＳＡの記号を運用することにより、計算時間が７０％より小さい値にまで節減され得ることは、明らかなことである。

ρ領域からの検索のために効果的な方法

ここでは、前述のセクションのべき乗処理および乗算処理のプロトコルが改善され得る。さらに、繰り返し動作がなされる間に、新たに前もって計算された定数Ｔを取り入れることにより、ρ領域内での乗算の回数を減らすことが可能になる。この場合、Ｔは、モジュロＮおよび指数Ｅの関数である。このような方法を、下記の処理過程に示す。

【００８６】

【数２８】

$$T=(2^{s})^{\Sigma}\ \mathrm{mod}N=(I^{-1})^{\Sigma}\ \mathrm{mod}N$$

ここで、$\Sigma=2q^{-1}+E \bmod 2q^{-1}$

また、

ｑは、Ｅに関連するビットの数を示す（最後部の０の部分は無視）

【００８７】この場合、モジュラ・べき乗は、下記の手順にて遂行される。

【００８８】

【数２９】

27

```
初期条件：   B＝A

             FOR j＝2 TO q
                B≒ρ（B・B）N

                IF  E（i）＝1  THEN
                   B≒ρ（B・A）N
             END  FOR

             B≒ρ（B・T）N
```

28

＊【００８９】ここで、各ステップから次のステップに移行する際に、ＢがＮに等しいかまたは大きい場合には、常に、ＢからＮが引かれることを再び仮定する。さらに、ρ領域内でのすべての乗算は、同じ因子Ｉによるモジュラ・乗算（例えば、ρ（Ｘ・Ｙ）＝Ｘ・Ｙ・Ｉ ｍｏｄＮ）に相当する点に再び注意すべきである。

【００９０】例３

この例３は、$\overline{A^E}$ ｍｏｄＮの計算におけるＴの有用性を証明し、かつ、Ｔの定義を明らかにするためのものである。この例３の処理過程を下記に示す。

【００９１】

＊ 【数３０】

ｎ＝４およびＥ＝５＝０１０１₂と仮定し、さらに、ｑ（Ｅの最後部の０を無視した後）を３に設定すると、

Ｅ（１）＝１；Ｅ（２）＝０；ａｎｄＥ（３）＝１，

次のように、Ｔが前もって計算される

$$T=(2^{s})^{\Sigma}\ \mathrm{mod}N=(I^{-1})^{\Sigma}\ \mathrm{mod}N$$

$$\Sigma=2q^{-1}+E\ \mathrm{mod}\ 2q^{-1}=2^{s-1}+5\ \mathrm{mod}\ 2^{s-1}$$

$$=4+1=5$$

ゆえに、

$$T=I^{-5}\ \mathrm{mod}N$$

【００９２】さらに、下記の処理過程を遂行する。 ※【数３１】

【００９３】 ※

```
初期条件：
        B＝A
  j＝2，E（２）＝０
        B≡ρ（B・B）N＝A²・ImodN
  j＝3，E（３）＝１
        B≡ρ（B・B）N≡B²≡A⁴・I²・ImodN
        B≡ρ（B・A）N≡A⁴・I³・A・ImodN
  最終的に、
        B≒ρ（B・T）N＝A⁵・I⁴・I⁻⁵・ImodN
                ≡A⁵ modN
```

【００９４】A^E を計算するために、次のようなステップが続けて実行される場合には、パラメータＴの導入が回避され得る。ここで、我々は、前もって計算されたモントゴメリの定数が存在すると仮定し、かつ、本発明の装置が、Ｐ領域内で２乗および乗算の両方の処理を遂行すると仮定することにより、下記の計算を実行する。

【００９５】$C=A^E\ \mathrm{mod}N$

この場合、Ｅ（ｊ）は、指数Ｅに対し２進法のビット表示を行ったときのｊ番目のビットを示す。このｊ番目のビットは、インデックスが１の最上位のビットで始まり、インデックスがｑの最下位のビットで終わる。奇数の指数に対しては、次のような処理過程によりべき乗を遂行することができる。

【００９６】

【数３２】

29

```
A* ¥ρ (A・H) N
B=A*
FORj=2TOq-1
      B¥ρ (B・B) N
      IF  E(j)=1  THEN
            B¥ρ (B・A*) N
ENDFOR
B¥ρ (B・A) N
C=B
```

30

＊【００９７】各ステップから次のステップに移行する際に、ＢがＮに等しいかまたは大きい場合には、常に、ＢからＮが引かれる。最後の繰り返し動作の後に、Ｂの値は、$A^E \bmod N$に対し限定された合同の関係（Ｂ￥$A^E \bmod N$）を有するようになる。そして、最終値としてＣが得られる。

【００９８】また一方で、偶数の指数に対しては、前述の最後のステップが下記の式(33)によって置き換えられる。

【００９９】

【数３３】

＊

B¥ρ (B・I) N（B¥ρ (B・A) Nに取って代わる）　(33)

【０１００】さらに、ここでの処理過程をより明確にするために、下記のような具体例を掲載する。

※【０１０１】

※　【数３４】

E＝1011→E(1)＝1；E(2)＝0；E(3)＝1；E(4)＝1；

$A^{1011} \bmod N$を求める；q＝4

A＝ρ (A・H) N＝$AI^{-8}I = AI^{-1} \bmod N$

B＝A

for j＝2 to q

B＝ρ (B・B) Nより$A^2 (I^{-1})^2 \cdot I = A^2 \cdot I^{-1}$が導き出される

E(2)＝0；	B＝$A^2 \cdot I^{-1}$	
j＝3	B＝ρ (B・B) N＝$A^2 (I^{-1})^2 \cdot I$	
	＝$A^4 \cdot I^{-1}$	
E(3)＝1	B＝ρ (B・A*) N	
	＝$(A^4 \cdot I^{-1})(AI^{-1}) \cdot I = A^5 \cdot I^{-1}$	
j＝4	B＝ρ (B・B) N＝$A^{10} \cdot I^{-2} \cdot I = A^{10} \cdot I^{-1}$	

E(4)が奇数なので、最後の乗算はAに基づいて遂行される。

この結果、寄生関数I^{-1}が除去される

B＝ρ (B・A)＝$A^{10} \cdot I^{-1} \cdot A \cdot I = A^{11}$

C＝B

【０１０２】Ｈパラメータの計算

Ｈパラメータは、モントゴメリの領域内での計算に際し不可欠なものであり、かつ、一定の値である。ある種のプロトコルを用いた場合、Ｈは、比較的大きなコンピュータで予め計算されるような定数になる。あるいは、他のプロトコルを用いた場合、Ｈは、より有効な定数を計算する際に使用される第１段階のパラメータであるような有用性のある定数にもなり得る。この点に関しては、前述のセクションを参照されたい。★

★【０１０３】通常の通信においては、Ｈが前もって計算されることを仮定している。しかしながら、幾つかのプロトコル、例えば、ＲＳＡにおけるランダムな通信の際の記号の暗号化に対しては、本発明の装置、例えば、ＳＭＡＲＴカードを使用してＨを計算することも必要である。Ｈパラメータは、下記の式(35)により計算される。

【０１０４】

【数３５】

$H = 2^{2n} \bmod N$　(35)

【０１０５】この式(35)は、Ｈパラメータが通常の除算動作の剰余であることを意味する。この場合、最上位の１ビットと、これに続く最下位のビット値０の２ｎビット（２ｎ＋１ビット長のオペランド）とから構成される

31

ビット列が、モジュールの基数Nにより除算されることになる。ビット値1の1ビット、および、ビット値0のビット列からなる被除数に対し除数Nによる2進法の除算を遂行することは、Nによる順次の試行減算を実行することに相当する。すなわち、上記の除算処理は、最上位のn＋1ビットがNよりも大きい場合に、残りの試行過程の被除数からNを引くことに相当するものである（下記の例を参照のこと）。

【0106】この場合、元の被除数は2n＋1ビット長であるが、除算処理により生成される残りの試行過程の＊10

32

＊被除数は、明らかに、n＋1ビット長を超えることはない。さらに、最下位のけたが0になることも明らかである。例えば、次のような例を挙げる。すなわち、N＝1 1_{10}＝1011$_n$（したがって、Nのビット長は、4になる、したがって、n＝4）のときに、Hを求める例を挙げる。

【0107】除算の基数を2として、長い除算を手動により実行する。

【0108】

【数36】

```
          1 0111
      ------------
1011  | 1 0000 0000
        1011        減算成功
        0101 0      ←1回目の丸めの結果
        101 1       減算失敗
        101 00      ←2回目の丸めの結果
        10 11       減算成功
        10 010      ←3回目の丸めの結果
        1 011       減算成功
        0 1110      ←4回目の丸めの結果
          1011      減算成功
```

5回目（(n＋1)回目）の丸めの結果⇒0011＝H（10進数の3＝剰余）

【0109】最終的に、H＝3_{10}であることが確認された。上記の除算処理では、n＋1回の試行減算が存在する。さらに、試行減算により得られる被除数もまた、n＋1ビット長になることに注意すべきである。このような減算処理の手順は、後述の本発明のハードウェアの説明箇所で詳しく述べることとする。

【0110】

【課題を解決するための手段、および、作用】本発明は、大きな数に対しモジュラ・乗算およびモジュラ・べき乗を遂行するための超小形電子系装置に係り、適切なクロック手段および制御手段を有する標準のマイクロプロセッサに対するコンパクトな同期式の電子系超小形周辺機器からなる。

【0111】さらに、本発明この超小形電子系装置は、各々が細分化される共に、切替制御可能であり、かつ、前記クロック手段により制御される複数種のシフトレジスタと、多重化され、かつ、直列／並列形の2つのみのマルチプレクサと、ボロー検出器と、補助的な減算器および加算器と、ディレイ・レジスタおよび切替素子とを備えている。

【0112】このような超小形電子系装置は、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗を同時処理かつ同期方式により遂行するために、前述のすべての構成部品を集積化して形成する。好ましくは、本発明の超小形電子系装置は、ハードウェアの乗算、2乗およびべき乗に対し設計されたモントゴメリの方法をもとに展開されるような新奇かつ複合形で同期式のハードウェア装置により実現される。

【0113】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、並列動作方式に直列動作方式を取り入れた多数の同時処理と直列処理との複合形、すなわち、乗算、減算、加算、記憶形ディレイおよび2^kによる除算を遂行する装置として機能する。さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、かつ、膨大な内部バスの使用を回避し得る。

【0114】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき乗のための多数の直列処理を遂行し、一般の1μm技術を用いたSMARTカード用のI.SO7816の標準規格により規定されるマイクロチップ上に形成される程度に充分コンパクトである。

【0115】さらに、好ましくは、本発明の超小形電子系装置は、モントゴメリの方法を展開することにより、モジュラ・乗算、モジュラ・2乗およびモジュラ・べき

乗のための多数の直列処理を遂行し、基本のアーキテクチュアを変えることなく、特に、デュアルポート・アクセスのためのメモリを再設計することなく、かつ、ファームウェアの要求が少ない状態で、１つの内部バスを備えた任意のマイクロプロセッサにより制御することが可能である。

【０１１６】さらに、好ましくは、本発明の超小形電子系装置は、マイクロプロセッサを使用してカスケード形のρ領域内での２乗および乗算の処理手順を規定する。さらに、前記超小形電子系装置は、ｎビット長のシフトレジスタを含み、かつ、モジュラ・乗算、モジュラ・２乗およびモジュラ・べき乗を遂行する多重化部を備えている。この多重化部内に指数Ｅを記憶することが不要であるために、この多重化部による制御を簡単にし、また一方で、ほんのわずかな付加的なマイクロコントローラのＲＯＭコードしか必要としない。

【０１１７】さらに、好ましくは、本発明の超小形電子系装置は、Ｂのレジスタが回転動作を行っている間に、オンザフライ方式により２乗の被乗数を用いてA_iのレジスタをロードする結果として、前記A_iのレジスタを前記Ｂのレジスタにより再ロードする際に、マイクロコントローラによりＢおよび／またはＢ－Ｎの前回の計算処理の最終値が取り出されるおそれが回避される。

【０１１８】このために、このマイクロコントローラのＲＡＭが節減され、かつ、２乗の繰り返し動作の各々において少なくともｎクロック分の実効的なクロック・サイクルを除去することが可能になる。本発明の構成によれば、$Z／2^k$がＮよりも大きいか、またはＮに等しいかを決定し、たった１回の直列形の減算のみ行うような比較的小さいＮオペランドを達成することにより、モントゴメリの方法による簡単な装置から、２つの記憶用レジスタおよび独立の直列形の減算処理が除去されると共に、$Z／2^k－N$に関する単一の直列形の検出を行うことが可能になる。

【０１１９】さらに、本発明の構成によれば、３つの同時乗算処理を遂行する際に２つの直列／並列形の乗算器のみが使用されるように、半並列形式で回路の同期をとっている。このために、シリコンを用いた装置において全シリコン領域に対し直列／並列形の乗算器の占める面積の割合が４０％に抑えられる。

【０１２０】さらに、本発明の構成によれば、ｋビットのシフトレジスタからなる１つのディジタルのディレイ素子を使用してＸの直列形の加算と乗算器の直列結果との同期をとので、直列／並列形の乗算器の積または繰り返し処理が二重に記憶されることが防止される。さらに、本発明の好ましい実施態様においては、シフトレジスタが、ｎビット長またはｎ／２ビット長で構成され、ｎ／２の長さのモジュールに対するべき乗が、ｎビット長のべき乗に対し必要であろうと思われる実効的なクロック・サイクル期間の１／８より幾分少ない時間で遂行される。

【０１２１】さらに、本発明の好ましい実施態様においては、オンザフライ方式によりＡのレジスタをロードし、かつ、オンザフライ方式によりＳのレジスタの内容の大きさを予測し、さらに、オンザフライ方式により一部のオペランドの同期をとることにより、ｎビットの数の乗算処理ρ（Ａ：Ｂ）Ｎが、実効的なｍ（ｎ＋２ｋ）クロック・サイクルで完全に遂行される。

【０１２２】さらに、本発明の好ましい実施態様においては、小規模のボロー検出回路が付加され、かつ、制御メカニズムに簡単な付加物が付加されているようなモントゴメリの乗算処理に対し使用されるものと同じ機器の同じレジスタを用い、第２のモードにおいてＨパラメータの計算を行うことができる。本発明のモジュラ・乗算を遂行するための方法においては、被乗数Ａ、乗数ＢおよびモジュロＮの各々が、ｋビット長のｍキャラクタから構成され、乗数ＢはモジュロＮよりも大きくない値に設定される。

【０１２３】前記の方法は、下記のステップにより実行される。第１のステップで、Ｈパラメータと、他のパラメータの少なくとも最下位のキャラクタJ_0とを前もって計算し、かつ、このキャラクタJ_0をｋビットのレジスタにロードし、第２のステップで、前記乗数ＢおよびモジュロＮを、それぞれ対応するｎビット長のレジスタにロードし、ここで、ｎ＝ｍ・ｋのように表され、第３のステップで、ｎビット長のレジスタＳのビット値をすべて０にし、第４のステップで、ｉ番目の繰り返し動作をｍ回遂行し、ここで、ｉは０からｍ－１までの数であり、さらに、ｉ番目の繰り返し動作の各々は、以下の動作を含み、（ａ）前記被乗数Ａのｉ番目のキャラクタA_iを、A_iのレジスタ手段から、レジスタおよびラッチ手段から選定された記憶手段へ転送し、（ｂ）Ｘ＝Ｓ（ｉ－１）＋Ａ（ｉ－１）＊Ｂにより表されるＸの値を生成し、ここで、Ｓ（ｉ－１）はＳの更新された値であり、Ｓの更新は、次のように定義され、

①乗算手段に対し、Ｂのレジスタを周期的に右方向へシフトし、

②直列形式でＢをA_iにより乗算し、

③前記モジュロＮを周期的に右方向へシフトし、

④Ｓ（ｉ－１）がＮよりも大きくない場合、（ｉ－１）番目の繰り返し動作の後にＳのレジスタに記憶される値をＳ（ｉ－１）の更新された値として決定し、Ｓ（ｉ－１）がＮよりも大きい場合、直列形式でＳ（ｉ－１）からＮを引くことにより得られる値をＳ（ｉ－１）の更新された値として決定し、さらに、この結果として得られるＳ（ｉ－１）の更新された値を設定し、

⑤Ｓのレジスタを周期的に右方向へシフトし、さらに、各ビット毎に、乗算Ａ（ｉ－１）＊ＢをＳ（ｉ－１）の更新された値に加算し、

（ｃ）Ｘ（X_0）の最下位のキャラクタをJ_0により乗

算し、NおよびXがkクロック・サイクルだけ遅延されている間に、$X_0 * J_0 \bmod 2^k$ の値をY_0のレジスタ手段に入れ、（d）$Z=X+Y_0 * N$のZの値を計算し、この計算は、次のように行われ、

①Nのレジスタに対し遅延かつ右方向へのシフトがなされた状態でY_0をNにより乗算し、同時に、この乗算結果に対し、前述の周期的な右方向へのシフトがなされ、

②Xを$Y_0 * N$の値に加算し、

（e）Zの最下位のキャラクタを無視し、残りのキャラクタをSのレジスタに入れ、このときに、最後の繰り返し動作以外は、$Z/2^k$ を入れることになり、（f）前述と同様の方法によりS（i－1）の更新された値を決定するために、各ビット毎に$Z/2^k$ とNとを比較し、

（g）前記被乗数Aのi番目のキャラクタA_iを、前記の動作期間において、Aのレジスタ手段にロードし、第5のステップで、最後（m回目）の繰り返し動作においては、$Z/2^k$ の最下位のキャラクタを無視し、残りのキャラクタを、C¥ρ（A＊B）NとしてBのレジスタに入れ、第6のステップで、前記第3および第4のステップを繰り返し、ここで、CがNよりも大きい場合には、CまたはC－NがBにとって代わり、さらに、HがAにとって代わることによってP＝ρ（C＊H）Nを計算し、第7のステップで、最後の繰り返し動作により得られるPの値を、A＊BmodNと仮定する。

【0124】さらに、本発明の方法によれば、被乗数Aと乗数Bとが同じ数である場合に、モジュラ・2乗およびモジュラ・乗算が遂行される。さらに、本発明の方法によれば、$D=A^E \bmod N$により表されるモジュラ・乗算およびモジュラ・べき乗が遂行される。さらに、本発明の方法は、

1．モジュラスをレジスタNに格納する工程、

2．レジスタSを0にセットする工程、

3．べき乗化されるべきベースAをレジスタBに格納する工程、

4．べき指数Eをコンピューターのレジスタに格納する工程、

5．該べき指数Eを左にシフトさせる工程、

6．第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、

7．該ビットのそれぞれに付いて、0か1かに関係なく、上記で定義された乗算方法により該レジスタBの内容を二乗すると同時に、該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、

8．若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後該レジスタBの内容を該ベースAで乗算する工程、及び

9．べき指数Eの全てのビットに付いて、工程6～8の操作が実行された後に、D¥＝A^E modNとしての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている。

【0125】さらに、本発明の方法は、

1．モジュラスをレジスタNに格納する工程、

2．レジスタSを0にセットする工程、

3．べき乗化されるべきベースAをレジスタBに格納する工程、

4．べき指数Eをコンピューターのレジスタに格納すると共に、以下に定義する事前演算パラメータTをコンピューターCPUに格納する工程、

記載の方法。

【0126】5．該べき指数Eを左にシフトさせる工程、

6．第1番目の1ビットに先行するそれらの0ビットを無視すると共に、該べき指数Eと第7及び8の操作を実行する為のビットに続く全てに対して、第1番目の1ビットを無視する工程、

7．該ビットのそれぞれに付いて、0か1かに関係なく、請求項1で定義された乗算方法に関して工程4及び5を実行すると同時に、被乗数と乗数とがベースAであり、且つ該ベースの連続する特性値が該レジスタBからレジスタAに格納される工程、

8．若し、べき指数Eに関する現在のビットが、1であるか、或いは1に過ぎない場合に、工程7の操作終了後、請求項1で定義された乗算方法に関して工程4及び5を実行し、その際、該被乗数がレジスタBの内容であり、且つ該乗数がベースAである工程、及び

9．べき指数Eの全てのビットに付いて、工程7～8の操作が実行された後に、レジスタBの内容を前記パラメータで付加的に乗算し、TD＝A^E modNとしての最後の操作に付いての結果を該レジスタBに格納する工程とから構成されている事を特徴とする請求項19による繰り返し操作を実行することにより、モジュラ・べき乗D¥＝A^E modNを実行する。

【0127】さらに、本発明の方法は、コンピューターCPUと乗算回路を含む制御手段から構成されている請求項19記載の方法により、モジュラ・乗算を実行する装置であって、該乗算回路は、乗数としてのn－ビットシフトレジスタB、モジュラスとしてのn－ビットシフトレジスタN、本発明に於いて定義されている値Sとしてのn－ビットシフトレジスタN、被乗数としてのk－ビットシフトレジスターAi、本発明に於いて定義されている値J0及びY0としてのk－ビットレジスタ手段、該レジスタBの内容と該レジスタAiの内容とを掛け合わせる乗算手段、付加的なn－ビット乗算器手段及び加算、減算、多重化及び遅延手段とを含んでいる。

【0128】さらに、本発明の方法は、該n－ビットレジスタとその他の構成部分との間の接続、及びラッチ回路以外の構成部分間の接続は1ビット接続である。さらに、本発明の方法は、

1．べき指数Ｅをコンピューターの記憶手段に格納する工程、
2．モジュラスを前記レジスタＮに格納する工程、
3．前記レジスタＳを０に設定する工程、
4．前記した特許出願番号１０４７５３に記載された方法に従って、Ａ＊＝ρ（ＡＨ）$_N$ の乗算操作を実行する工程、（此処で、Ａは、べき乗化されるべきオペランドでありＨは、前記で定義した事前演算パラメータである。）
5．該Ａ＊を該ベースレジスタＢに格納する工程、
6．該ベースレジスタＢの内容に対して二乗演算操作を実行する工程、
7．該べき指数Ｅを左にシフトさせる工程、
8．第１番目の１ビットに先行するそれらの０ビットを無視すると共に、該べき指数Ｅと第９及び１０の操作を実行する為のビットに続く全てに対して、第１番目の１ビットを無視する工程、
9．該ビットＥのそれぞれに付いて、０か１かに関係なく、上記で定義された二乗方法による工程４及び５の操作を実行する工程であって、該被乗数と該乗数とが共に該レジスタＢから派生されるものであり、且つ該モントゴメリー乗算器に於ける連続する特性値が該レジスタＢからレジスタＡｉに格納される工程、
１０．若し、べき指数Ｅに関する現在のビットが、１であるか、或いは１に過ぎない場合に、工程９の操作終了後、前記で定義された二乗方法に関して工程４及び５を実行し、その際、該被乗数がレジスタＢの内容であり、且つ該乗数がベースＡ＊である工程、及び
１１．べき指数Ｅの全てのビットに付いて、工程８～１０の操作が実行された後に、レジスタＢの内容を前記オリジナルベースＡで付加的に乗算し、Ｄ￥＝A^EｍｏｄＮとしての最後の操作に付いての結果を該レジスタＢに格納する工程とから構成されている。

【０１２９】さらに、本発明の方法は、平均有効長がｎ／２ビットである２つの数値に付いて従来の乗算を実行する方法で有って、該乗算方法は、該請求項１９に於いて定義されている乗算方法により該数値に対してモジュラ・乗算処理を実行するもので有って、該モジュラ、Ｎ、は全てが“１’ｓ”（fffffff.....fff）で構成されたｎ－ビット数で、Ｊ０から１に対応するものであり、被乗数をレジスタＢに格納し且つ請求項１に於いて定義されている乗算方法に従ってＡを取り扱うものであり、Ｎは、全て１によりプリローディングレジスタＮの手段によるか、或いは一連の“ハード”１を出力する為のＮを出力するマルチプレクサーをセットすることにより、該Ｎは全て１と成りうるものである。

【０１３０】

【実施例】本発明のモジュラ・乗算およびモジュラ・べき乗を遂行する超小形電子系装置ならびにその遂行方法は、添付図面（図１～図９）を参照しながら、本発明の好ましい実施例を具体的に説明することにより、より良く理解されるであろう。これらの添付図面は、本発明の装置を全体的に理解するために必要な複数種の論理的概念を示すものである。すべての場合において、クロック信号に従い回路が動作する。そして、リセット信号がある場合には、このリセット信号は、回路を零の状態にすることを目的としている。

【０１３１】以下、図１～図９の添付図面を参照しながら、本発明の実施例を詳細に説明する。図１は、本発明の一実施例の装置構成を示すブロック図である。ここでは、本発明の装置が集積化されたモノリシック回路のブロック図が例示されている。図１において、多重化部（ＭＵＬＴ部と記載されることもある）は、本発明の基礎となるハードウェア装置を備える。状態マシンは、多重化部の回路を駆動するための制御部を構成する。ＲＯＭ（読み出し専用回路）部は、すべて不揮発性メモリ（ＲＯＭおよびＥＰＲＯＭ）から構成される。このＲＯＭ部には、ＳＭＡＲＴカードを制御するためのプログラム、高信頼性の３つのグループからなる公開キー、および、多重化部や状態マシンを駆動するためのプログラムが格納されている。ＲＡＭ（ランダムアクセス回路）部は、一時的に存在するオペランドを記憶するための揮発性メモリから構成される。この種のオペランドとして、べき乗処理がなされる予定のメッセージ、暗号化されるべき公開キー、多重化部に転送されるべきデータ等が挙げられる。ＣＰＵ（中央処理装置）は、実際は、８ビットまたはそれより大きなビットの内部バスを有するような任意のマイクロコントローラである。

【０１３２】図２は、本発明の一実施例のモジュラ・乗算回路を示すブロック図である。この場合、モジュラ・乗算回路は、モジュラ・２乗およびモジュラ・べき乗を実行するために用いられる。図２において、参照番号１０、１１および１２は、Ｂ、ＳおよびＮのレジスタをそれぞれ構成するようなｎビット長（ｎ＝ｋ・ｍ）の３つのレジスタを示すものである。これらのレジスタの各々には、乗数の値Ｓとモジュロの値がロードされる。上記レジスタは、好ましくは、２つのｎ／２レジスタに分割されている。さらに、上記レジスタは、好ましくは、ＮおよびＢのレジスタに対し、ｋビットの最下位ビット部分を含む。マルチプレクサ１３、１４および１５は、それぞれ、上記レジスタの前部に配置される。この場合、もし、これらのマルチプレクサが細分化されて個々の部品として形成されているならば、各マルチプレクサが、各々のレジスタ前部に配置される。さらに、図２のブロック図に示すように、３つのレジスタは、直列形式でロードされるように意図されている。しかしながら、並列形式によるロードも可能である。

【０１３３】１６、１７および１８は、いずれもｋビット長であり、かつ、A_i、J_0、およびY_0の値をそれぞれ受け入れる３つのレジスタを示すものである。レジ

スタ１６、１７は、それぞれ、直列ロード／並列出力形のシフトレジスタと、直列および並列ロード／並列出力形シフトレジスタである。レジスタ１８は、好ましくは、直列入力／並列出力形シフトレジスタである。これらのレジスタの内容は、それぞれ、構成要素２１、２２を介して乗算手段１９、２０により処理されるように意図されている。これらの構成要素２１、２２は、好ましくは、ｋビットのラッチである。これらの構成要素２１、２２がラッチである場合、これらの構成要素２１、２２は、ｋビットのバスを通してレジスタ１６、１７および１８からロードされる。また一方で、上記の構成要素２１、２２がレジスタである場合、これらの構成要素２１、２２は、１ビットの接続部を通して直列形式でロードされ得る。

【０１３４】参照番号２４、２５、２５′、２６、３６、３７および３８は、マルチプレクサを示している。乗算手段（乗算器）１９、２０は、直列入力Ａ、並列入力Ｂ、および直列出力を有する乗算手段か、または、その他の直列／並列入力、および直列出力を有する乗算手段である。マルチプレクサ３８は、通常の数の領域内で乗算処理を実行するために、モジュロＮのビット値を強制的にすべて１（オール１）にするものである。

【０１３５】参照番号２７、２８、２９、３０および３１は、１ビットの全加算／減算手段または半加算／減算手段を示している。この内、３１は、全加算／減算手段を示している。参照番号３２、３３および３４は、ディジタル信号を遅延させることが可能なｋビットかつｋクロック・サイクルのディレイ手段を示している。これらのディレイ手段は、アナログ素子またはディジタル素子のいずれによっても構成され得るが、アナログ素子により構成するのが好ましい。３５は、ボロー検出器を示している。このボロー検出器は、２ビットのラッチ／記憶手段である。図２からわかるように、本発明の装置は、例えば５１２ビットのような大きい数を取り扱うように意図されているにもかかわらず、わずかな数のｋビットのバスをオプションで持っている以外には、バスを備えていない。このために、本発明の装置では、ハードウェアの節減が図れる。Ｂ、ＳおよびＮのレジスタが、ｎ／２ビットの部分を有する場合、本発明の装置は、２５６ビットの数に対し乗算動作およびべき乗動作を遂行するために使用することが可能である。このために、本発明では、装置を使用する際の柔軟性を備えるという利点が生ずる。

【０１３６】図３は、本発明の一実施例の特殊なモジュラ・乗算回路を示すブロック図である。ここでは、本発明の一実施例のモジュラ・乗算回路を論理セルにより構成している。図３において、オペランドは、直列の接続部ＤＩを介し、A_i のラッチと、J_0 のレジスタと、Ｂのレジスタと、Ｎのレジスタとに供給される。そして、オペランドによる処理結果は、直列の接続部ＤＯを介し、ＢのレジスタまたはのNのレジスタから取り出される。

【０１３７】信号Ｘは、ＢとＡ$_i$ とＳとの積Ｂ・Ａ$_i$ ・Ｓのビットの流れを合計した結果（和）に相当する（ＳとＢの積は、Ｎよりも小さいと仮定する）。信号Ｙ$_0$ は、Ｊ$_0$ とＸの積Ｊ$_0$ ・Ｘにおける最下位のｋビットの流れに相当する。信号Ｚは、Ｙ$_0$ とＮの積Ｙ$_0$ ・Ｎと、Ｘとの和に相当する。ここでは、Ｚ中の最下位のｋビットはすべて０なので、この最下位のｋビットは無視される。この結果、最上位のｎビットのみが、ＳまたはＢに対し直列に供給される。

【０１３８】ボロー検出器は、Ｚ／2^k の値がＮより大きいか否かを検出するための論理回路である。減算器（通常、Ｓｕｂと略記される、図３では、単に減算と記す）１および減算器２は、ＢおよびＳの値がＮよりも大きい場合は、常に、ＢおよびＳのビットの流れからＮのビットの流れを減算するように動作する。

【０１３９】加算器（通常、Ａｄと略記される、図３では、単に加算と記す）１および加算器２は、Ｘの流れとＺの流れを生成するために、ビットの流れを加算するように動作する。ディレイ素子（図３では、単にディレイと記す）１およびディレイ素子２は、シフトレジスタから構成される。これらのディレイ素子は、数学的処理の同期をとるための記憶手段を提供するために必要なものである。

【０１４０】図３では、クロックの制御は図示していない。ここでは、クロックは、状態マシンから供給されると仮定している。このクロックの供給は、前述の直列入力／直列出力形の論理回路の１つからデータを送り出したりこれらの論理回路の１つにデータを提供したりしなければならないときは、いつでも行われる。他の制御、すなわち、マルチプレクサのアドレス制御や、ラッチ転送信号の制御等もまた、詳しく開示していない。なぜならば、これらの制御は、本明細書に含まれる説明事項より、当業者にとっては明らかであるからである。

【０１４１】さらに、図２および図３の装置が、どのようにして本発明の乗算の方法に関係する複数の動作を遂行するかは、当業者にとっては明らかであろう。しかしながら、これらの複数の動作のタイミング関係は、念のために、次の図４に示す。図４は、本発明の一実施例による繰り返し動作（イテレーション）と乗算動作との間の時間的な関係を示す図である。この図においては、本発明の一実施例による実効的かつ連続的なクロック・サイクルにおいて遂行されるようなすべての各種動作が、図式的に示されている。この場合、ｎ＝５１２、および、ｍ＝１６に設定される。このような設定条件は、暗号化技術においては、比較的一般的な条件である。前述の図３に例示された実施例に従って本発明を実施する場合、ｎ＝２５６の条件下で本発明を実行するために、図４と同じ装置が使用される。

【０１４２】図４においては、一連の各種動作が、実効的なクロック・サイクルの関数として図示されている。この実効的なクロック・サイクル（実効クロック）に関しては、横軸に目盛りがふられている。各種動作の始まり、および、すべての繰り返し動作前のタイミングでは、Ｂ、ＳおよびＮの値が、それぞれ対応するレジスタにロードされる。上記の繰り返し動作は、本発明の乗算の方法の一部をなす。Ａの最初のキャラクタもまた、対応するレジスタにロードされる。ｋクロック・サイクルの期間において繰り返し動作が始まるや否や、ＢおよびＳのレジスタの内容のシフト動作が行われる。ｎ＋ｋの実効的なクロック・サイクルの期間においてＸの値が発生する。最初のｋクロック・サイクルは、X_0 の値を取り入れることにより占有される。最初の実効的なｋクロック・サイクルの期間において、Y_0 の値が取り入れられる。次の実効的なｎ＋ｋクロック・サイクルの期間において、乗算器２０に既に取り入れられているＸの値がシフトするか、または、このＸの値がディレイ素子３４により遅延された後に加算器３１に取り入れられる。Ｎの値は、３つの異なる時間位相にて使用される。初めの位相は、ＳおよびＢを更新するために使用される。第２番目の位相は、実効的なｋクロック・サイクルの遅延期間の後に、Y_0 による乗算を遂行するために使用される。第２番目の位相は、２回目の実効的なｋクロック・サイクルの遅延期間の後に、ＳまたはＢの次の値がどのようにして更新されるかを検知するために使用される。同様の実効的なｎ＋ｋクロック・サイクルの期間において、Ｚが計算され、かつ、$Z／2^k$が計算される。最初の実効的なｋクロック・サイクルの始まりのタイミングで、A_i のロードが始まる。さらに、繰り返し動作が連続している間は、A_i のロードも続けて行われる。$Z／2^k$ の最終値は、最初の実効的な２ｋクロック・サイクルの期間後のｎクロック・サイクルの期間において、Ｓ（またはＢ）のレジスタに取り入れられる。

【０１４３】図５は、直列／並列乗算器のセルの構成を示す回路図である（この回路図の作成に際しては、専門の技術に精通している技術陣の助けを借りているが、彼らは、本発明に関係する直列／並列乗算器のセル構成の研究に関しては関与していない）。これらの複数のセルの各々は、後述の図６に示すような乗算器（通常、ＭＰＬと略記される）を備える。

【０１４４】図６は、８ビットの直列／並列乗算器の構成を示す回路図である。この直列／並列乗算器は、符号のない直列／並列乗算器の乗算動作に対しブース（Booth）の乗算アルゴリズムを実行する。図３の乗算器（通常、ＭＬと略記される、図３では、単に乗算と記す）１および乗算器２に示すような直列／並列乗算器は、ｋビット長である。この場合、ＭＳセル、すなわち、最上位ビットのセルが退化していることに注意すべきである。並列の８ビットの被乗数が、ＸＩの接続部に入力される。さらに、ｎビット長の直列の乗数がＹの接続部に入力される（乗数の最上位の１ビットの後に、最初に現れる最下位のｋビットの列はすべて０である）。さらに、乗算器による乗算結果である積は、出力側の接続部ＭＯにおいて、最下位のｋビットが最初に現れ、最上位のビットが最後に現れる。この場合、積の全体は、ｎ＋ｋビット長になる。

【０１４５】図７は、直列加算器の構成を示す回路図である。ここでは、Ａの接続部とＢの接続部に現れる２つのビットの流れを加算するための直列加算器が例示されている。この直列加算器においては、出力側の接続部Ｓにおいて、ビットの流れの和が出力される。図７においては、最下位のビットが最初に入力される。さらに、ｍビット長のオペランドに対する出力の流れは、ｍ＋１ビット長になる。ｍ回の実効的なクロック・サイクルの最後の部分では、ＣＩの出力は、数のビット列中のｍ＋１番目のビットに相当する。

【０１４６】図８は、直列減算器の構成を示す回路図である。ここでは、Ａの接続部とＢの接続部に現れる２つのビットの流れの差を出力するための直列減算器が例示されている。この直列減算器においては、出力側の接続部Ｄにおいて、ビットの流れの差が出力される。図７においては、最下位のビットが最初に入力される。さらに、ｍビット長のオペランドに対する出力の流れは、ｍビット長になる。ｍ回の実効的なクロック・サイクルの最後の部分では、ＢＩの出力は、数のビット列中のｍ＋１番目のビットに相当する。同様に、このＢＩの出力は、ボローを表示するためのボロー表示手段として機能する。

【０１４７】図９は、Ｈパラメータを計算するためのアーキテクチュアを示すブロック図である。ここでは、ｎビット長のモジュールＮに対しＨパラメータを計算するためのハードウェア構成が例示されている。このような動作モードの期間では、ｎビット長のモジュールに対し、Ｎのレジスタがｎ＋１回だけ回転動作を遂行する。この回転動作は、Ｓのレジスタの回転動作に同期した状態で行われる。この場合、Ｓのレジスタは、減算器１を介し、最下位のビットの遅延と一緒に回転動作を遂行する〔最下位のビット値０は、最初のクロック・サイクルにおいてマルチプレクサ（Ｍ２　１；１）に挿入される〕。ボロー検出器は、回転動作が完了するような最後のタイミングで、次回の丸めにおいてＳの流れからＮが引かれるか否かを認識する。さらに、ボロー検出器は、次回の丸めに応じて前回の減算マルチプレクサを切り替える。

【０１４８】前述のように、図１は、本発明の方法を遂行するための装置をブロック図の形で表したものである。図１の装置における制御部は、下記の構成要素を備える。

(1) 完備した形のＣＰＵ（中央処理装置）

(2) カウンタ
(3) 状態マシン
さらに、ＣＰＵは、不揮発性メモリおよび揮発性メモリを有する。これらの不揮発性メモリおよび揮発性メモリの幾つかは、乗算処理過程に使用され得る。さらにまた、上記ＣＰＵは、回路内のモジュールの計算機能ブロックを制御する。

【０１４９】さらに詳しくいえば、上記ＣＰＵは、下記の機能を有する。
(1) ホストと交信すること
(2) チップにデータをロードし、かつ、チップからデータを取り出すこと
(3) 回路に対し一連の数学的動作を遂行するように指示すること
(4) 他の暗号化システムおよび非暗号化システムに応答してデータ処理動作を遂行すること
カウンタは、実際の状態マシンに対するアドレスを生成する。

【０１５０】状態マシンは、アドレスを復号化し、多重化部（ＭＵＬＴ部）に対する複数種の制御信号を生成する。これらの制御信号は、多重化部に対し、ρ（Ａ・Ｂ）Ｎ変換の計算を実行するために必要とされる適切な動作手順を遂行するように指示する（ここで、ＡはＢに等しい）。図３は、本発明の物理的な形態（多重化部）を実施するためのハードウェアの装置をブロック図の形で表したものである。さらに、図３は、本発明に係る特許により保護されるべきアーキテクチュアの幾つかの概念に焦点を絞る際の補助手段となるように意図されている。図３のブロックは、同時に、モントゴメリのモジュラ・乗算にて既述したような式(1) ～(5) により規定された手順を遂行する。さらに、図３のブロックは、同期クロックを変えることなく、かつ、限定された合同の関係を有するＳおよびＢの変換を行うことなく上記の手順を遂行する。このセクションにおいて、我々は、定数（Ｎの関数）J_0 およびＨが前もって計算されることを仮定している。 図３の回路は、ρ（Ａ・Ｂ）Ｎを遂行する。この回路の機能を利用することにより、この回路は、次の計算を行うために使用され得る。

【０１５１】(1) Ｂ・ＡｍｏｄＮ
(2) B^2 ｍｏｄＮ
ただし、いかなる場合でも、ＢはＮよりも小さくなければならない。ここで、Ｃ＝Ｂ・ＡｍｏｄＮを遂行する手順を詳細に説明する。
(1) まず、プロセッサが、オペランドＢをＢのレジスタに前もってロードする。同様に、プロセッサは、オペランドＮをＮのレジスタに前もってロードする。

【０１５２】(2) 多重化部内の回路が次回のＳの値を計算し始める度に、回路は、次回のA_i を前もってロードすることを（フラグを立てることにより）ＣＰＵに知らせる。Ｓ（ｍ）回の繰り返し動作の後に、Ｂに対し限定された合同の関係を有する数がＢのレジスタに残る。
(3) 多重化部が、Ｆ＝ρ（Ｂ・Ｈ）Ｎを計算する。ここで、プロセッサがH_iのキャラクタの処理手順を前もってロードする場合を除けば、Ｈは、上記のステップ(1) および(2) に記載された手順において、前もって計算された定数である（プロセッサがA_i のキャラクタを前もってロードする場合も同じことがいえる）。

【０１５３】また一方で、Ｃ＝B^2 ｍｏｄＮを遂行する手順を詳細に説明する。
(1) まず、Ｂのレジスタが、Ｂに対し限定された合同の関係を有することがわかっている数を保持していると仮定する。さらに、Ｎのレジスタが、モジュールＮを保持していると仮定する（２乗処理においては、一般にいえることである）。ここで、多重化部は、B_0 とB_0 の最下位のキャラクタでもってA_i のレジスタを予めロードしておくことにより、２乗処理を進めることができる。

【０１５４】(2) Ｂ＝ρ（Ｂ・Ｂ）Ｎの計算処理は、上記の乗算動作における第２のステップ（ステップ(2)）と同様の処理過程で進行する。ただし、Ｂのレジスタが回転動作を行っている場合に、B_i のキャラクタの連続するローディング動作が、Ｂのレジスタから直列式かつオンザフライ方式になされるときは、この限りではない。

【０１５５】(3) 必要な場合には、上記の乗算動作における第３のステップ（ステップ(3)）と同様のρ（Ｂ・Ｈ）の計算を行う。当業者にとっては明らかなことなので、発明者は、直列／並列形乗算器および一般の構成部品が本発明そのものの一部を構成することは、敢えて主張しない。今後の説明は、一般に普及している標準の論理セルを使用していることを明確にするためになされるものである。ただし、論理セルの幾つかは、それほど度々一般に使用されていないかもしれない。ここで図示したゲート構成は、本発明の証明のために例示しているにすぎない。熟練された技術者は、これらの論理セルを最適化するであろう。

【０１５６】オペランドＡ、ＢおよびＮは、いずれもｎビット長であり、ｋビット長のキャラクタからなるｍ個のグループにより構成される。それゆえに、ｎ＝ｋ・ｍが成り立つ。ｋ＝３２のハードウェア装置においては、ｍは、８ビットまたは１６ビットの２進数のビット長になる。

乗算器１および乗算器２
これらの乗算器（ＭＬ）は、符号のない乗算動作に対しブースの乗算アルゴリズムを実行する。この場合、並列のオペランドは、ｋセル（ビット）長になっており、直列のロードされるオペランドは、任意の要望されるビット長になっている。

【０１５７】各々の直列／並列乗算器は、ｋ－１のＭＰＬセルからなる（図５参照）。ＭＳビットに相当する最上位のセルは、ＡＮＤゲートのみから構成される。各々

のＭＰＬセルは、Ｙの直列の入力ビットと、ＸＩの並列の入力ビットとの乗算動作を行う。さらに、この乗算動作により、前段のＭＰＬユニットの直列出力と、それ自身の前回のサイクルのキャリー出力ビットが生成される。上記のＭＰＬセルは、これらの出力結果を合計する。

【０１５８】図５からわかるように、各々のＭＰＬセルは、２ビットの乗算加算器である。ＭＰＬセルのブロックは、ＸＩの入力ビットと、Ｙの直列の入力ビットとの＊乗算動作を行う。さらに、このブロックは、ＤＩ（データ入力）からの乗算結果と、前回のサイクルからのＣＩ（キャリー入力）との加算処理を行う。最終の結果として、ＤＯ（データ出力）と、次のサイクルに対するＣＯ（キャリー出力）が得られる。このキャリー出力ＣＯは、Ｄフリップフロップに記憶される。データ出力ＤＯは、下記の式(37)により表される。

【０１５９】

【数３７】

$$DO=(DI+CI+XI\cdot Y)\bmod 2 \qquad (37)$$

【０１６０】このようにして記憶されたキャリー出力ＣＯは、次のサイクルに対するキャリー入力ＣＩになる。このキャリー出力ＣＯは、ブーリアン（Boolean ）の和により、下記の(38)により表される。

加算器１および加算器２

ここで使用される各々の加算器（Ａｄ）は、Ｄフリップフロップからなる単純な１ビットの全加算器である。この加算器は、次のクロック・サイクルで出力されるキャリービットを記憶するために使用される（図７参照）。

【０１６１】図７に示すように、２つの入力Ａ、Ｂは、前回のクロック・サイクルからのキャリー入力ＣＩと一緒に加算される。この結果、モジュロ２の和が生成される。この和は、出力信号Ｓを取り出すために、Ｄフリップフロップに記憶される。加算器がリセットされたときは、キャリービットは０になる。

減算器１、減算器２および減算器３

図８に示すように、減算器（Ｓｕｂ）の各ブロックは、前段のボローを記憶するためのＤフリップフロップからなる全減算器である。このブロックは、前述の加算器のブロックとほぼ同じ構成になっている。ただし、減算器においては、Ａの流れからＢの流れを直列に引く点が加算器と異なる。

【０１６２】ディレイ素子１、ディレイ素子２およびディレイ素子３

これらのディレイ素子（Ｄｅｌａｙ）は、ｋ１ビットの連結された状態の記憶素子から構成される。これらのディレイ素子は、数学的処理において各種のオペランドの同期をとるために使用される。これらの同期動作は、回路を説明すれば明らかになるであろう。

【０１６３】A_i、J_0、およびY_0

これらのブロックは、ｋビット長の直列入力／並列出力形シフトレジスタである。この場合、ｋビットの入力ビットは、直列に入力される。ｋビットの実効的なクロック・サイク期間の後に、これらのｋビットは、並列形式で出力側に現れる。

【０１６４】図２においては、細い線が直列の１ビットの導体線を示し、太い線が並列のｋビットの導体線を示している。

Ｍ４　１；ｘ、Ｍ３　１；ｘ、およびＭ２　１；ｘ

これらのブロックは、１ビット出力のマルチプレクサである。Ｍ４　１；ｘは、４つの入力から１つの出力を取り出すものである。Ｍ３　１；ｘは、３つの入力から１つの出力を取り出すものである。Ｍ２　１；ｘは、２つの入力から１つの出力を取り出すものである。ｘは、特定の構成部品に対する明瞭なインデックスを示している。

Ｂ（０：ｋ－１）、Ｂ（ｋ：ｎ１－１）、Ｂ（ｎ１：ｎ２）、Ｓ（０：ｎ１－１）、Ｓ（ｎ１：ｎ２）、Ｎ（０：ｋ－１）、Ｂ（ｋ：ｎ１－１）、およびＢ（ｎ１：ｎ２）

これらのブロックは、シフトレジスタである。比較的ビット長の長いレジスタのビット列の大きさおよび位置が、括弧（ ）内の数字によって示されている。例えば、Ｘ（ｓ：ｔ）は、ｔ－ｓ＋１ビット長のシフトレジスタである。ここで、ｓは、レジスタＸ（ｓ：ｔ）の最初のビットのインデックスであり、ｔは、レジスタＸ（ｓ：ｔ）の最後のビットのインデックスである。例えば、Ｂ（０：５１１）は、次のような３つの比較的短いカスケード接続形のレジスタから構成される。すなわち、Ｂ（０：３１）、Ｂ（３２：２５５）、およびＢ（２５６：５１１）から構成される。

【０１６５】ｎ１は、一般に、ｎ／２（例えば、２５６）に等しい。ｎ１は、ｋの倍数でなければならない。ｎ２は、ｎ－１に等しい。ｋは機器のキャラクタの長さ、すなわち、直列／並列乗算器の大きさである。したがって、最初の処理過程においては、次の値が予測される。

【０１６６】ｎ１＝２５６、ｎ２＝５１１、ｎ＝５１２、およびｋ＝３２

ラッチ１およびラッチ２

これらの２つのラッチは、ｋビットのレジスタである。これらのラッチは、乗算器内の並列データを保持するために使用される。このようなラッチの動作により、乗算処理において単一のクロックを用いた並列変換が可能になる。

【０１６７】多重化部（ＭＵＬＴ部）の動作…ρ領域内での乗算およびべき乗

説明を簡単にするために、我々は、レジスタ内のデータが実際に移動するクロック・サイクルのみを指定することとする。すなわち、我々は、このようにデータが移動

するサイクルを実効的なクロック・サイクルと定義する。

ρ（A・B）Nの乗算

第1段階：初期ローディング

この段階では、DIを介して下記のレジスタがロードされる。

【0168】(1) J_0 のレジスタにJ_0 をロード（CPUにより前もって計算される）

(2) Bのレジスタに Bをロード

(3) Nのレジスタに Nをロード

(4) A_2 のレジスタに A、A_0 の最初のキャラクタをロード

同時に、ステップ(2) において、レジスタSに対しビット値0がロードされる。

【0169】これらの5つのレジスタに所定の値をロードした後に、符号のな2つの直列／並列乗算器ML1およびML2と、直列加算器Ad1およびAd2と、直列減算器Sub1、Sub2およびSub3とがリセットされる。

第2段階：B・A_0 の繰り返し動作の実行

レジスタA_i にロードされたデータA_0 は、ラッチ1に転送される。レジスタBは、周期的に右方向へのシフト動作を行う。繰り返し動作の始まりにおいては、ボロー検出器2の制御信号はビット値0になっている。それゆえに、Bの内容は、Sub1を通過しても変化しないままでいる。さらに、Bの内容は、ML1においてA_0 により乗算される。レジスタBの出力は、変化しない状態で、その入力される。

【0170】このような乗算の結果は、Ad1において、レジスタSの内容に直列に加算される。最初の繰り返し動作のときは、レジスタSの内容はすべて0である。この動作により、前述のようにXが生成される。上記の処理過程が進行している間に、CPUは、Aの次のキャラクタであるA_1 をラッチ1に予めロードしておく。

【0171】さらに、J_0 のレジスタからラッチ1にJ_0 がロードされる。XがML2に直列に入力され、J_0 により乗算される。実効的なkクロック・サイクルが経過した後に、レジスタY_0 の内容は、積$X_0 \cdot J_0$ の最下位のkビットになる。さらに、最初の実効的なkクロック・サイクルが経過した後に、ML2はリセットされる。ここで、直列入力形のマルチプレクサ M3 1；4は、Xの流れをNの流れに切り替える。レジスタY_0 内のデータは、J_0 に取って代わり、ラッチ2に並列にロードされる。さらに、ラッチ2の出力は、$Y_0 \cdot N$の流れに切り替えられる。次のn＋kクロック・サイクル期間においては、ML2からの直列の出力結果は、$Y_0 \cdot N$になる。実効的なkクロック・サイクル期間だけ遅延されたXは、今度は、Ad2において加算処理がなされ、ML2の積を生成する。この結果、$Z = X + Y_0 \cdot N$が得られる。ここで、Zは、最下位のkビットがすべて0であるような数である。

【0172】Ad2の最初のkビットはすべて0なので、この最初のkビットは無視される。そして、次のnビットが、Sのレジスタに直列に戻される。繰り返し動作の最終的な値は、Nに等しいかまたは大きい（この場合には、この値をNから引くことが必要である）。すなわち、S（1）￥S（1）modNが成立する。SがNよりも大きいか否かを検出するために、Sub3においては、nビット長の（Z／2^n）の流れからNが減算される。しかしながら、この場合、n回目のボロービットのみが、ボロー保持用フリップフロップに記憶される。

【0173】もし、このボロービットのビット値が0であるか、または、Ad2の最終キャリービットCOのビット値が1であれば、Sのレジスタの最新の値はNよりも大きい。最初の繰り返し動作の始まりにおいては、S（1）modNに対し限定された合同の関係を有する数がSのレジスタ内に存在する。J_0 、BおよびNのレジスタは、最初にロードされたオリジナルな値を保持する。そして、データを前もって保持するためのA_i のレジスタはA_1 を保持する。

【0174】第3段階：その後のB・A_i の繰り返し動作の実行

Aの次のキャラクタであるA_1 が、ラッチ1およびML1の並列入力に転送される。次のB・A_i の繰り返し動作およびそれに続く繰り返し動作の期間中、各々の繰り返し動作の最後において、S（i）modNに対し限定された合同の関係を有する数Sが存在する。もし、S（i）がNよりも大きければ、Sub2においてS（i）からNが引かれる。

【0175】各々の繰り返し動作が始まるときに、CPUは、Aの次のキャラクタであるA_1 を、データを前もって保持するためのA_i のレジスタにロードしておく。

ρ（B・B）Nの2乗動作

通常のべき乗処理の最初の動作は、2乗動作である。この2乗動作は、Bのレジスタにロードされた乗数Aと、A_i のレジスタにロードされた被乗数との通常の乗算と同じような手順で行われる。ただし、この場合、ビット数は、前述したようにkビット分だけ増加する。さらにその後の2乗動作は、Bのレジスタ内に存在するような限定された合同の関係を有するオペランド（乗数および被乗数）により遂行される。

【0176】上記のρ（B・B）Nのような2乗動作が遂行されている間、J_0 、S、BおよびNの各レジスタは、その出力側で、前回の乗算および2乗処理により得られた値を変えることなくそのままロードされる。しかしながら、この場合は、繰り返し動作において、A_i のレジスタは、Bのレジスタ内に存在するkビットのキャラクタから派生する新しいキャラクタをロードしなければならない。

【０１７７】上記の連続的な２乗動作において、A_i のレジスタは、Ｂのレジスタからオンザフライ式に予めロードされる。ＣＰＵが、一度、２乗処理を遂行するように指示を与えると、それ以降の２乗動作は問題なく遂行される。Ｂのレジスタにロードされる$B(i)'$は、Ｓｕｂ１を通って流れるＢの一部分である（B_i の中で、既にＮよりも小さい部分）

第１段階：$B \cdot B_0$ の繰り返し動作

先ず、前回の計算処理から導き出されるような、Ｓに対し限定された合同の関係を有する最新の数が、Ｂのレジスタ内に存在するものとする。

【０１７８】レジスタＢ、Ｎの最下位のｋビットは、周期的に右方向へのシフト動作を行う。さらに、実効的なｋクロック・サイクルの後に、レジスタＢ、Ｎは、オリジナルの状態に復帰する。レジスタＢ内の値は、適当なＢの値か、または、次のρ領域での乗算を遂行するために使用されるＢ－Ｎの値である。したがって、最初の丸めにおいて、レジスタA_i は、B_0 またはＢ－Ｎの最下位のｋビットでもって予めロードしなければならない。ここで、B_0 は、レジスタＢ内に存在する値である。

【０１７９】この最初のｋビットの回転動作の目的はレジスタＡｉへのプリロードの最初のｋビットがＳｕｂ１を通って流れることを可能にするためである。直列にロードされた直後に、ＡｉはLatch １にアンロードされ、Ａｉプリロードレジスタはの第２の文字であるB_1 をロードするために自由にされる。このおよび引き続く操作の間、Borrow２信号がセットあるいはリセットされた時にSub １からの出力は正であり、常にＮより小である。

【０１８０】全ての値がレジスタにロードされると、説明されるようにＢがローテイトした時にB_1 がＡｉレジスタ中にロードされる点（乗算においてＣＰＵはＡｉレジスタにロードすることを思い出すこと。）を除くと、以前に説明したようにこの最初の乗算は$B \cdot A_0$ に対して実行される。第２のｋビット文字B_1 はＢストリームから発生するために、この最初の$B \cdot B_0$ 処理の間にB_1 セグメントは、次の開平演算、すなわち$B \cdot B_1$ 繰り返しのために直列的にＡｉプリロードレジスタ中にオンザフライ処理で切り換えられる。

第２段階：$B \cdot B_1$ 繰り返し動作

A_i のレジスタ中にロードされた値B_1 は出力ラッチLatch １に転送される。次のｎ＋２ｋ（すなわちｎ＋６４）クロックサイクル中に$B \cdot B_1$ に対する乗算処理が上述のように実行される。

【０１８１】前回と同様Borrow１およびBorrow２信号は、ＢおよびＳレジスタから発生する流れからＮが減算されうるか否かを決定する。もしＳレジスタ中の値がＮより大であるかあるいは等しければBorrow１はセットされ、減算器Sub １においてＮはＳから減算される。もし必要であればｍ繰り返し乗算ループ完了の間にＮはＳから減算される。このような状況は先行する乗算あるいは開平演算の終わりにBorrow２で検知される。

【０１８２】フリップフロップBorrow１およびBorrow２はSub ３からの条件付きのボロウ出力の最終値を記憶している。Borrow１は各Ｓの繰り返しの後にセットあるいはリセットされる。Borrow２はＢがＳ（ｍ）にロードされる最後のＳ（ｍ）繰り返しの後にセットあるいはリセットされる。この条件付きのボロウ出力はＳ（ｉ）がＮより大であるか否かを示す信号である。

【０１８３】$B \cdot B_1$ 処理の間、文字B_2 が減算器Sub １中に存在する場合に文字B_2 はオンザフライ処理でＡｉプリロードレジスタ中にロードされる。

第３段階：次の$B \cdot B_1$ 乗算繰り返し

文字B_1 が減算器Sub １中に存在する時には文字B_1 の値がＡｉレジスタにロードされる間に残りのｍ－２回の繰り返しが次のループの準備のために実行される。

【０１８４】限定された一致の最終結果はＳおよびＢレジスタ中に存在する。このデータはＤＯを通して直列に出力されるために、もし必要であればSub １で修正されるであろう。

乗算ブロックの操作－Ｈパラメータの演算

Ｈを演算するために、マシンは図９に示されるようにレジスタＳおよびＮを使用するために再構成される。上記で既に使用した数値例を用いて演算子の操作を説明する。この構成はＨの演算をｎ＋１回で実行する。各回の実行においてＳおよびＮは共にローテイトされ、各ローテイトはｎクロックである。各実行回においてＮは回転し変化せずに帰還する。ｉ回目の実行において、Ｓおよび次の減算（Next Subtract）信号はＳ（ｉ）の限定された¥一致の同等値を含んでいる。

初期条件－第１回実行

第１回目の実行の最初で、ＮはＮレジスタ中にロードされ、第１回目の試行減算が成功したことを表すボロー検出フラグはリセットされ、Sub １の出力フリップフロップはゼロにリセットされる。第１回目の実行中、試行除算のｎ番目のＭＳビットは“１”である。このビットは次の減算用フリップフロップ（Ｓ中にスペースはない。）を推論して記憶される。次の減算は、第１回目の実行においてＳ－Ｎ減算を命令する。上述のｎ＝４ビットの数値例を使用して例証する。

51　　52

H計算モード－初期条件
　ボロー検出器の次減算フラグに格納される
　　↓
まず、被除数のＭＳビットが　　N＝１０１１₂，ｎ＝４
”１”であることを知る。

図７参照

それ故、ボローが存在し得ない
ことを知るので、次の減算フラグを
零にリセットする。　　Ｓ（０）　Ｓレジスタの内容
　　↓　　↓
－－－－－－－－－－－－　（０）０００〔０００〕←”仮想零”
　　↓　　↑↑↑↑
ボロー検出次減算　　〔これら”仮想”のＬＳ零は
信号は零であるーそれ故、最初の　　試用減算によって影響されない。各ラウンドで、”仮想零
ラウンドでＭ２＿１：３はＮをＳｕｂ１　　カウンタ”に１－０が存在するだろう。〕
に与えるー差はリーディング零を
伴うＳ－Ｎか、ちょうど２・（Ｓ－Ｎ）
となろう。
最初のクロックサイクルで、リセット
Ｓｕｂ１出力フリップフロップからの
零は、ＳからのＬＳビットがＳｕｂ１
に与えられるとき、ＳのＭＳセルに
与えられる。
（ＳのＬＳビットは常に”仮想”ＬＳ
零カウンタ”から”引かれる”零で
である。）
最初のｎ－１クロックサイクルの間、
ＤｉｆｆのＬＳのｎ－１ビットが
Ｓに与えられる。
ＮはそのＭＳビットセルに回転バックされる。
ＢＯ（ボローアウト）シリアルストリームは
　Ｄｉｆｆ　ｍｏｄ　２ⁿ－Ｎストリーム
から結果するボローの一続きに等しい。しかし、
最後のボローのみサンプルされ、関連する
かもしれない。
第ｎの実効クロックサイクルにて、
ＤｉｆｆのＭＳビットが”１”であるか又は
ＢＯ＝”０”ならば、”次減算”が次の
ラウンドの減算用にフラグを上げる。
最初のラウンドにて２ⁿ からＮがひかれ、２（ＬＳ零挿入）を乗じられた結果の
ｎビットはＳレジスタにリターンされる。ただし、”推論によって”ボロー検出
次減算レジスタに格納されるＭＳビットを除く。
最初のラウンドの終わりにて回転する：
Ｓ（１）＝１０１０、次の減算＝１（ＢＯ＝１）、そして、次のラウンドでＳｕ
ｂ１にはＳ－Ｎの減算が存在しない。

53　　54

Hパラメータ計算－第２ラウンド
ボロー検出の次の減算フラグに格納される

まず、第２ラウンドの減算が、
Ｓｕｂ２にて”検出された”
ＢＯ＝”１”として成功しないだろう
ことを知る。

Ｎ＝１０１１，，ｎ＝４

Ｓ（１）　最初のラウンド後のＳレジスタの内容

（１）　１０１０　〔０００〕←”３仮想零”が残る

ボロー検出次減算
信号は１－それ故このラウンドで
Ｍ２＿１；３はＳｕｂ１に零を
与えよう－Ｄｉｆｆ＝２・Ｓ
減算が存在しなかった
（ＳのＬＳビットは再び”仮想”
ＬＳ零カウンタ”から”引かれた”
零である。）
続くｎ－１クロックサイクルの間
Ｄｉｆｆ＝２・ＳのＬＳのｎ－１
ビットがＳレジスタに与えられる。
NはそのＭＳビットセルに回転バックされる。
ＤｉｆｆのＭＳビットが”１”のとき、
次のラウンドでＳ－Ｎを減算しなければ
ならないことを知る。
サンプルされたＢＯは不適切である。
Ｄｉｆｆ＝１　０１００かつＳ（２）＝０１００　そして、次のラウンドでＳｕｂ１にてＳ－Ｎの減算があることを知る。

【０１８６】

Hパラメータ計算－第３ラウンド
ボロー検出器の次減算フラグに格納される

まず、ＤｉｆｆのＭＳビットが”１”だったので、第３ラウンドの減算が成功するだろうことを知る。

Ｎ＝１０１１₂，ｎ＝４

Ｓ（２）　第２ラウンド後のＳレジスタの内容

（０）　０１００　｛００｝←”２仮想零”が残る

ボロー検出次減算
信号は零。ＤｉｆｆからＮが引かれる。
続くｎ－１クロックサイクルの間、Ｄｉｆｆ＝２（Ｓ－Ｎ）のＬＳのｎ－１ビットがＳレジスタに逆供給される。
ＤｉｆｆのＭＳビットが次のラウンドのＳｕｂ１において”１”なので、Ｓ－Ｎの減算をしなければならない。
Ｄｉｆｆ＝１　００１０　かつ　Ｓ（３）＝００１０、次の減算＝０　かつ次のラウンドでＳｕｂ１においてＳ－Ｎの減算があることを知る。

【０１８７】

Hパラメータ計算－第４ラウンド
ボロー検出器の次減算フラグに格納される

まず、ＤｉｆｆのＭＳビットが”１”だったので、第４ラウンド減算が成功するだろうことを知る。

Ｎ＝１０１１₂，ｎ＝４

Ｓ（３）　第３ラウンド後のＳレジスタの内容

（０）００１０｛０｝←”１仮想零”が残る

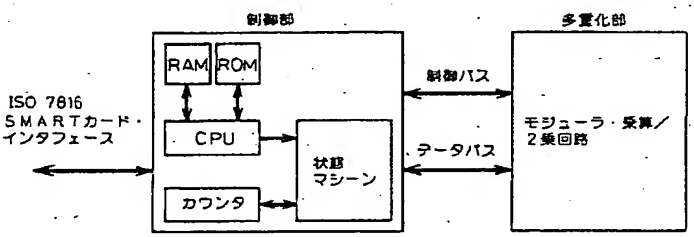
ボロー検出次減算
信号は零。ＮがＤｉｆｆから引かれる。
ボローＢＯ＝”０”がなかったので、次のラウンドでＳ－Ｎの減算をする。
Ｄｉｆｆ＝０　１１１０　かつ　Ｓ（４）＝１１１０，次減算＝０，そして、次のラウンドではＳｕｂ１においてＳ－Ｎの減算があろうことを知る。

【０１８８】

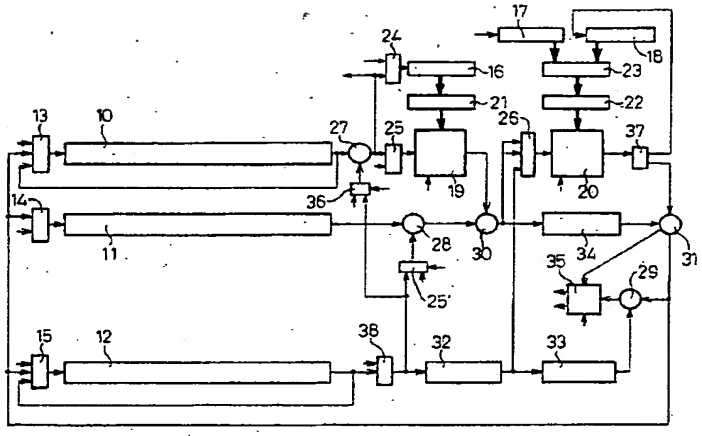
Hパラメータ計算－第ｎ＋１（第５）ラウンド
ボロー検出器の次減算フラグに格納される
↓
まず、ＤｉｆｆのＭＳビットが”１”だったので、第４ラウンド減算が成功するだろうことを知る。
Ｎ＝１０１１₂，ｎ＝４
Ｓ（４）　第４ラウンド後のＳレジスタの内容
↓　↓
（０）１１１０｛｝←”ノー仮想零”が残る最終ラウンド
↓
ボロー検出次減算
信号は零。ＮがＤｉｆｆから引かれる。
Ｄｉｆｆ＝０　００１１　かつ　Ｓ（５）＝００１１₂が残りである。それはＨの値である。

【図面の簡単な説明】
【図１】本発明の一実施例の装置構成を示すブロック図である。
【図２】本発明の一実施例のモジューラ・乗算回路を示すブロック図である。
【図３】本発明の一実施例の特殊なモジューラ・乗算回路を示すブロック図である。
【図４】本発明の一実施例による繰り返し動作と乗算動作との間の時間的な関係を示す図である。
【図５】直列／並列乗算器のセルの構成を示す回路図である。
【図６】８ビットの直列／並列乗算器の構成を示す回路図である。
【図７】直列加算器の構成を示す回路図である。
【図８】直列減算器の構成を示す回路図である。
【図９】Ｈパラメータを計算するためのアーキテクチュアを示すブロック図である。
【符号の説明】
１０～１２…レジスタ
１３～１５…マルチプレクサ
１６～１８…レジスタ
２７～３１…加算／減算手段
３２～３４…ディレイ手段
３５…ボロー検出器

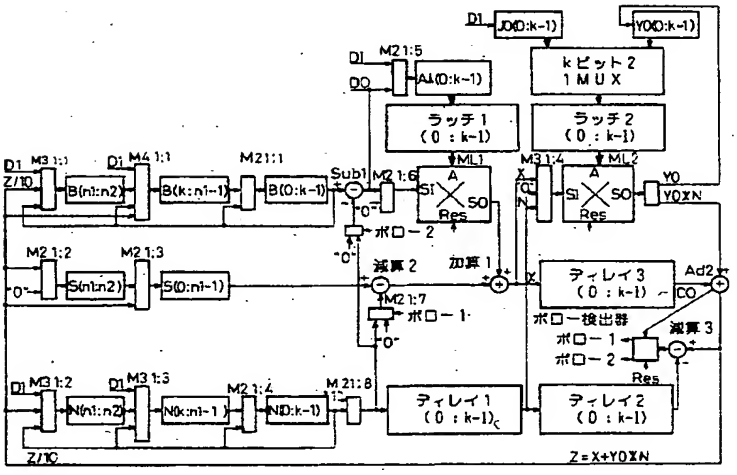
【図１】



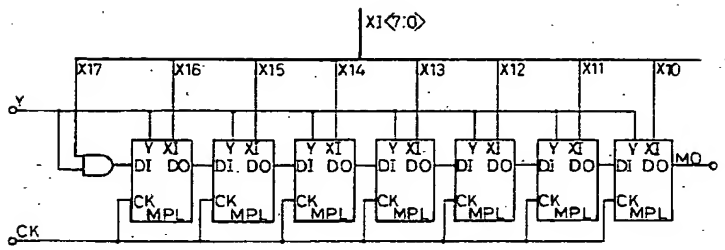
【図２】


17
24
18
16
23
21
22
13
10
27
25
26
37
19
20
14
36
11
28
30
34
31
35
29
25'
15
12
38
32
33

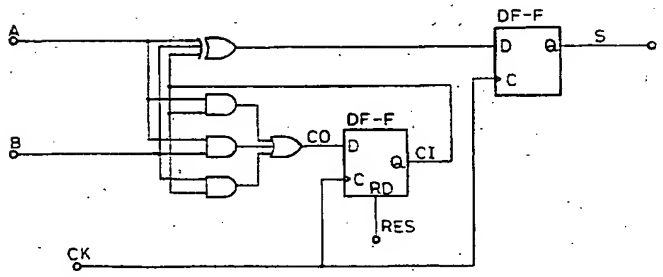
【図３】


D1
J0(0:k-1)
Y0(0:k-1)
D1
M2 1:5
D0
A1(0:k-1)
kビット2
1MUX
ラッチ1
(0:k-1)
ラッチ2
(0:k-1)
D1
M3 1:1
Z/10
B(n1:n2)
D1
M4 1:1
B(k:n1-1)
M21:1
B(0:k-1)
Sub1
M2 1:6
ML1
ML2
M3 1:4
A
SI
SO
Res
Y0
Y0XN
ボロー2
M2 1:2
M2 1:3
S(n1:n2)
S(0:n1-1)
減算2
加算1
ディレイ3
(0:k-1)
CO
Ad2
M2 1:7
ボロー1
ボロー検出器
減算3
D1
M3 1:2
D1
M3 1:3
N(n1:n2)
N0k:n1-1)
M2 1:4
N(0:k-1)
M21:8
ディレイ1
(0:k-1)
ディレイ2
(0:k-1)
Z/10
Z=X+Y0XN

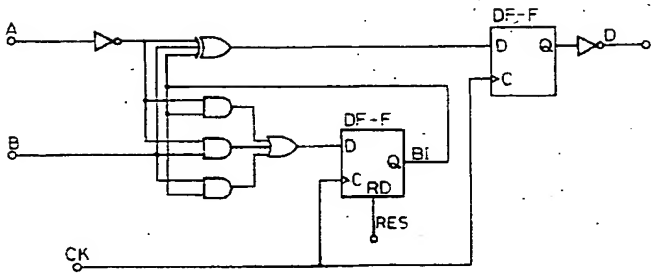
【図6】


XI<7:0>
X17
X16
X15
X14
X13
X12
X11
X10
Y
Y XI
DI DO
CK MPL
MO
CK

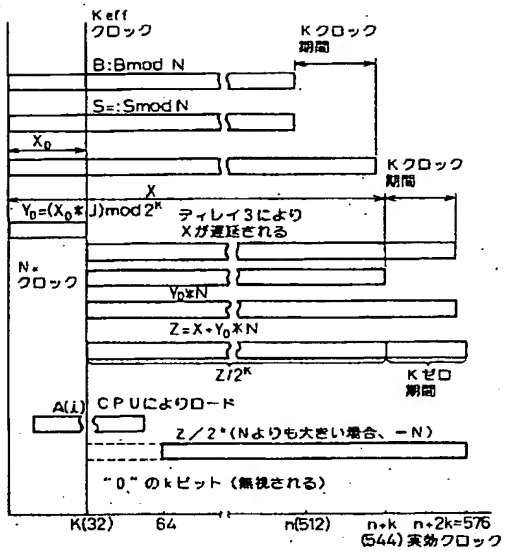
【図7】


A
B
CK
DF-F
D
Q
C
S
CO
CI
RD
RES

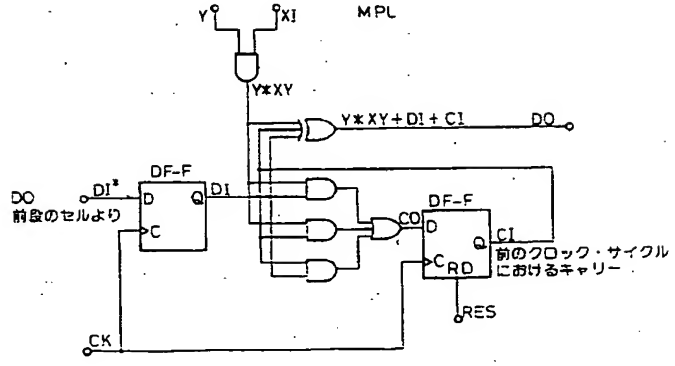
【図8】


A
B
CK
DF-F
D
Q
C
D
BI
RD
RES

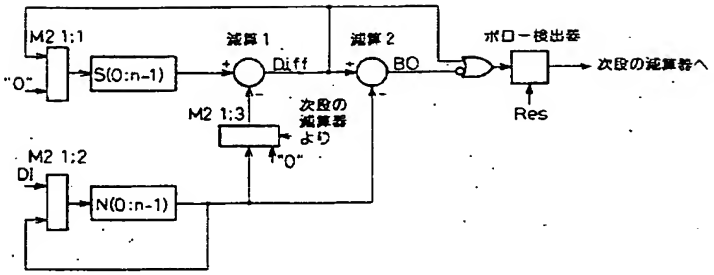
【図4】


Keff
クロック
Kクロック
期間
B:Bmod N
S=:Smod N
X0
Kクロック
期間
X
Y0=(X0*J)mod 2K
ディレイ3により
Xが遅延される
N*
クロック
Y0*N
Z=X+Y0*N
Z/2K
Kゼロ
期間
A(i)
CPUによりロード
Z／2*(Nよりも大きい場合、－N)
"0" のkビット（無視される）
K(32)
64
n(512)
n+k
n+2k=576
(544)実効クロック

【図5】


Y
XI
MPL
Y*XY
Y*XY+DI+CI
DO
DF-F
DO
前段のセルより
DI*
D
Q
DI
C
CO
DF-F
D
Q
CI
C
RD
前のクロック・サイクル
におけるキャリー
RES
CK

【図９】



フロントページの続き

(72)発明者　イタイ　ドローア
イスラエル国，ビアーシェバ，ミブツァナチション　76／32

(72)発明者　イザーク　ハダッド
イスラエル国，ビアーシェバ　84434，デレチ　ハシャロム　105／3

(72)発明者　ベンジャミン　アラジ
イスラエル国，オマー　84965，シガロン　ストリート　38